

PlanAhead User Guide

UG632 (v 11.4) December 1, 2009



Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU "AS-IS" WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2009 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Included in the PlanAhead™ software source code is source code for the following programs:

Centerpoint XML

The initial developer of the Original Code is CenterPoint - Connective Software Engineering GmbH. Portions created by CenterPoint - Connective Software Engineering GmbH. Copyright © Copyright IBM Corp. 1998 1998-2000 CenterPoint - Connective Software Engineering GmbH. All Rights Reserved. Source Code for CenterPoint is available at <http://www.cpointc.com/XML/>

NLView Schematic Engine

Copyright © Copyright IBM Corp. 1998 Concept Engineering.

Static Timing Engine by Parallax Software Inc.

Copyright © Copyright IBM Corp. 1998 Parallax Software Inc.

Java Standard Edition

Copyright © Copyright IBM Corp. 1998 1995 - 2006 Sun Microsystems

Includes portions of software from RSA Security, Inc. and some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>.

Powered By JIDE - <http://www.jidesoft.com>

The BSD License for the JGoodies Looks

Copyright © Copyright IBM Corp. 1998 2001-2008 JGoodies Karsten Lentzsch. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of JGoodies Karsten Lentzsch nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Libconfig (v1.3.2) License

libconfig - A library for processing structured configuration files

Copyright (C) 2005-2009 Mark A Lindner

This file is part of libconfig.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, see <http://www.gnu.org/licenses/>.

About this Guide

The *PlanAhead User Guide* provides detailed information about the PlanAhead™ software, including an interface overview, and instructions for using the design and software capabilities.

This chapter contains the following sections:

- “About PlanAhead”
- “Guide Contents”
- “Additional Resources”
- “Conventions”

Note: For information on software installation, and system requirements, refer to the *Xilinx ISE Design Suite: Installation, Licensing, and Release Notes*.

About PlanAhead

PlanAhead is a design and analysis software product used to design FPGA devices. It provides an integrated and intuitive environment for the entire FPGA implementation process. With PlanAhead, you can realize circuit performance improvements by analyzing the design RTL sources, synthesized netlists and implementation results. You can experiment with different implementation options, refine timing constraints and apply physical constraints and floorplanning techniques to help improve design results. Early estimates of resource utilization, interconnect delay and routing connectivity can assist with appropriate logic design, device selection and floorplanning. A hierarchical database also enables a block-based incremental design methodology that can reduce the run times and compute resources required to place and route the design.

PlanAhead Features

- I/O pin planning environment
 - Interactive and automatic I/O placement
 - Group related I/Os into interfaces
 - Robust Design Rule Checks (DRCs)
 - Simultaneous Switching Noise (SSN) Analysis
 - Weighted Average Simultaneous Switching Output (WASSO) Analysis
 - Alternate device compatibility
- RTL development and analysis environment
 - RTL source (Verilog and VHDL) importing
 - RTL elaboration with construct checking

- RTL Editor with crossprobing
 - Resource estimation
 - RTL Netlist and Hierarchy views
 - RTL Schematic
 - RTL Design Rule Checks (DRCs)
- Synthesis and implementation environment
 - Configure and launch one or multiple runs
 - Define and apply commonly used ‘strategies’ to ‘runs’
 - Monitor and manage results
 - Import and analyze results
 - Multiple CPU capabilities on Linux
 - Linux remote host
- Design analysis
 - Interactive schematic viewer
 - Graphical hierarchy viewer
 - Netlist tree viewing and expansion
 - Resource utilization statistics (before Place and Route)
 - Advanced searching for device or design objects
 - Design metrics display
 - Object properties to display relevant information
 - Design rule checking
 - Connectivity display
 - Placement analysis and highlighting
- Static timing estimation (TimeAhead)
- Timing constraints editor
- Robust floorplanning environment
 - Hierarchical floorplanning
 - Flexible capabilities to assign and fix LOC constraints
- Block-based capabilities
 - Block level implementation
 - IP creation and reuse

Integration with the Project Navigator Environment

The PlanAhead software provides an environment to help improve your design results throughout the design flow. Integrated within the Project Navigator environment, PlanAhead is automatically launched at four different design process steps. These include I/O pin planning (Pre-Synthesis), I/O pin planning (Post-Synthesis), Floorplan Area/IO/Logic (Post-Synthesis) and Analyze Timing/Floorplan Design (Post-Implementation). Each of these steps offers unique and powerful capabilities previously only available in the standalone PlanAhead environment. The PlanAhead software replaces PACE and Floorplanner for all pin planning, design viewing, and floorplanning flows for FPGA designs.

When PlanAhead is invoked from Project Navigator, the interface provides access to only the PlanAhead features specific to the selected task. This mode of PlanAhead is called ISE® Integration mode. The mode is displayed in the status bar at the bottom of the PlanAhead viewing environment.

Guide Contents

This document contains the following chapters:

- [Chapter 1, “Understanding the PlanAhead Design Flow”](#) provides an overview of the benefits of using PlanAhead, an overview of the design flow and features in PlanAhead, and describes the required inputs and outputs to the product.
- [Chapter 2, “Creating and Managing Projects”](#) describes the initial setup and management of a project within PlanAhead.
- [Chapter 3, “Using PlanAhead With Project Navigator”](#) describes the PlanAhead flows that are integrated with Project Navigator.
- [Chapter 4, “Using the Viewing Environment”](#) describes the PlanAhead user interface.
- [Chapter 5, “I/O Pin Planning”](#) describes the PinAhead environment which enables pin assignment.
- [Chapter 6, “Creating and Analyzing the RTL Design”](#) describes the PlanAhead RTL environment.
- [Chapter 7, “Implementing a Design”](#) describes the PlanAhead synthesis and implementation environment.
- [Chapter 8, “Analyzing the Design”](#) describes the PlanAhead design analysis capabilities.
- [Chapter 9, “Analyzing Implementation Results”](#) describes the timing and placement analysis capabilities available in PlanAhead.
- [Chapter 10, “Floorplanning the Design”](#) describes how to use the product to create floorplanning constraints.
- [Chapter 11, “Debugging the Design with ChipScope”](#) describes the ChipScope software debugging capabilities now integration into PlanAhead.

This document contains the following appendices:

- [Appendix A, “Menu and Toolbar Commands”](#) provides a brief description of menu and toolbar commands.
- [Appendix B, “Installing Releases with XilinxUpdate”](#) describes the PlanAhead release strategy and explains how to update the software.
- [Appendix C, “Configuring SSH Without Password Prompting”](#) describes how to setup a passwordless SSH, which is required for running PlanAhead processes on multiple hosts.

Additional Resources

The following additional resources are available. For more information, go to the Xilinx[®] website (<http://www.xilinx.com/planahead>).

To find additional documentation, see the Xilinx website at:

<http://www.xilinx.com/support/documentation/index.htm>.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a webcase with Technical Support, see the Xilinx website at:

<http://www.xilinx.com/support/mysupport.htm>.

Xilinx Customer Education Training

- *Designing with PlanAhead*—Attend this Xilinx Customer Education Training Course to learn about the PlanAhead functionality using a sample design.

Documentation

- *Xilinx ISE Design Suite: Installation, Licensing, and Release Notes*—This document provides specific installation instructions and requirements. Available from the software and from the Xilinx website.
- *PlanAhead Release Notes*—The Release Notes provide specific information about new features in this release. Available from the software and from the Xilinx website.
- *PlanAhead Methodology Guide*—This Guide provides information about various strategies aimed at improving performance, repeatability of results or reducing design times. Available from the software and from the Xilinx website.
- PlanAhead Tutorials—The following tutorials are available with the PlanAhead software and on the Xilinx website:
 - *Quick Front to Back Flow Overview*
 - *I/O Pin Planning*
 - *RTL Development and Analysis*
 - *Design Analysis and Floorplanning*
 - *Debugging with PlanAhead and ChipScope*
 - *Using PlanAhead with Project Navigator*

Video Demonstrations

- *PlanAhead Technical Video Demonstrations*—Watch the video demonstrations to learn more about specific areas of the PlanAhead software. Available from the Xilinx website only at

http://www.xilinx.com/products/design_resources/design_tool/resources/index.htm

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
Courier font	Messages, prompts, and program files that the system displays	speed grade: - 100
Courier bold	Literal commands that you enter in a syntactical statement	ngdbuild <design_name>
Helvetica bold	Commands that you select from a menu	File > Open
	Keyboard shortcuts	Ctrl+C
Italic font	Variables in a syntax statement for which you must supply values	ngdbuild <design_name>
	References to other manuals	See the <i>PlanAhead Methodology Guide</i> for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are <i>not</i> connected.
Square brackets []	An optional entry or parameter. However, in bus specifications, such as bus [7:0] , they are required.	ngdbuild [option_name] <design_name>
Braces {}	A list of items from which you must choose one or more	lowpwr = {on off}
Vertical bar	Separates items in a list of choices	lowpwr = {on off}
Vertical ellipsis . . .	Repetitive material that has been omitted	IOB #1: Name = QOUT' IOB #2: Name = CLKIN' . . .
Horizontal ellipsis ...	Repetitive material that has been omitted	allow block block_name loc1 loc2 ... locn;

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section “ Additional Resources ” for details. Refer to “ Title Formats ” in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the <i>XST User Guide</i> .
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Preface: About this Guide

About PlanAhead	5
PlanAhead Features	5
Integration with the Project Navigator Environment	6
Guide Contents	7
Additional Resources	8
Xilinx Customer Education Training	8
Documentation	8
Video Demonstrations	8
Conventions	9
Typographical	9
Online Document	10

Chapter 1: Understanding the PlanAhead Design Flow

PlanAhead Design Flows	19
Basic Design Flow	20
Run Experimentation Flow	20
Design Analysis and Floorplanning Flow	21
Input and Output Files	21
Inputs to PlanAhead	21
Outputs for Reports	24
Outputs for Environment Defaults	26
Outputs for Project Data	27
Outputs for ISE Implementation	28
PlanAhead Terminology	32

Chapter 2: Creating and Managing Projects

Invoking PlanAhead	35
Linux	35
Windows	35
PlanAhead Command Line Options	36
Using a PlanAhead Startup Tcl Script	37
Using the Getting Started Jump Page	38
Understanding the Different Types of PlanAhead Projects	38
Empty Projects for I/O Pin Planning	38
RTL Source Based Projects	38
Synthesized Netlist Based Projects	38
Implemented Design Results Based Projects	39
Using the Create New Project Wizard to Create a New Project	39
Entering a Project Name and Storage Location for the Project	40
Selecting the Design Source Data Type	40
Creating an Empty Project for I/O Pin Planning	41
Creating a Project by Importing RTL Sources	43
Creating a Project with Synthesized EDIF or NGC Format Netlists	47
Creating a Project with ISE Placement and Timing Results	52
Managing Projects	59
Opening an Existing Project	59
Opening Multiple Projects	60
Closing a Project	61

Updating the Netlist for a Synthesized Netlist Based Project	61
Working with Floorplans	65
What is a Floorplan?	65
Creating a Floorplan	65
Importing Constraints	68
Importing Module-Level Constraints	69
Updating Design Constraints for Floorplans	70
Creating Multiple Floorplans	72
Managing Multiple Floorplans	72
Viewing or Editing Floorplan Properties	73
Closing a Floorplan	75
Deleting a Floorplan	76
Saving a Floorplan	76
Copying a Floorplan	77
Renaming a Floorplan	77

Chapter 3: Using PlanAhead With Project Navigator

Integration Overview	79
ISE and PlanAhead Integration Process	80
Passing Logic and Constraints	80
I/O Pin Planning (Pre-Synthesis)	81
I/O Pin Planning (Post-Synthesis)	81
Floorplan Area/IO/Logic (Post Synthesis)	82
Analyze Timing/Floorplan Design (Post Implementation)	82
PlanAhead Viewing Environments	83
I/O Pin Planning View Layout	83
Design Analysis and Floorplanning View Layout	85
Transitioning from PACE / Floorplan Editor to PlanAhead	86
Overview	86
Launch and Invocation of PlanAhead from Project Navigator	86
Launch and Invocation of Standalone PlanAhead	87
Empty Design & Pre-Synthesis I/O Pin Planning	87
Post Synthesis Pin Planning and Area based Floorplanning	88
General I/O Pin Planning and Floorplanning Tasks	89
Viewing ISE Implementation Results	90
Summary	90
Transitioning from Floorplanner to PlanAhead	90
Launch and Invocation of PlanAhead from Project Navigator	91
Launch and Invocation of Standalone PlanAhead	91
General Floorplanning Tasks	91
Timing Analysis - Path Visualization	92
Viewing ISE Implementation Results	93
Summary	93

Chapter 4: Using the Viewing Environment

The Viewing Environment	95
PinAhead Environment View Layout	96
Project Environment View Layout	97
Floorplan Environment View Layout	98
Navigating Views	99

Toggling Views Displayed	99
Opening Views	99
Manipulating Views using the View Banner Commands	100
Using the View Auto-Hide Capability	100
Floating Views	101
Defining Viewing Area Sizes	101
Using the View Specific Toolbar Commands	101
Using the Workspace Views	102
What are the Workspace Views?	102
Opening Workspace Views	103
Viewing the Workspace Full Screen	103
Floating the Workspace View	103
Printing the Workspace View	103
Closing Workspace Views	103
Splitting the Workspace	104
Using Common Environment Views	105
Using the Console View and Tcl Command Line	105
Using the World View	106
Using the Status Bar	107
Understanding Object Selection Options	108
Selecting Objects	108
Fitting the Display to Show Selected Objects	109
Using the Selection View	109
Highlighting Selected Objects	110
Marking Selected Objects	110
Setting Selection Rules	110
Setting Selection Ability for Objects in the Workspace Views	111
Understanding the Context Sensitive Cursor	111
Configuring the Viewing Environment	112
Customizing PlanAhead Display Options	112
Setting PlanAhead Behavior Options	118
Moving Views within Viewing Areas	121
Creating Custom View Layouts	122
Restoring a View Layout	123

Chapter 5: I/O Pin Planning

Recommended Method for Pin Planning With Xilinx FPGAs	125
PinAhead Overview	127
Using the PinAhead Environment	127
Using the I/O Ports View	129
Using the Package Pins View	130
Using the Package View	131
Using the Device View	132
Viewing Device Resources	132
Viewing Package Pin Properties	132
Viewing I/O Bank Resources	133
Viewing Clock Region Resources	134
Defining Alternate Compatible Parts	134
Importing I/O Ports	136
CSV Format File	136
HDL Format Files	137

UCF Format File	138
Defining and Configuring I/O Ports	138
Creating I/O Ports	138
Configuring I/O Ports	139
Prohibiting I/O Pins and I/O Banks	140
Creating I/O Port Interfaces	141
Placing I/O Ports	142
Enabling Interactive Design Rule Checking	143
Placing I/O Ports into I/O Banks	143
Placing I/O Ports in a Defined Area	144
Sequentially Placing I/O Ports	145
Automatically Assigning I/O Ports	146
Placing Gigabit Transceiver I/O Ports	148
Placing I/O Related Clock Logic	148
Removing I/O Placement Constraints	149
Configuring DCI_CASCADE Constraints	149
Editing a DCI Cascade	151
Running I/O Port and Clock Logic Related DRCs	151
Running Simultaneous Switching Noise (SSN) Analysis	157
Running SSN Analysis (Virtex-6)	157
Viewing the SSN Results	158
Resolving SSN Issues	159
Viewing I/O Bank Properties	160
Defining the I/O Port Switching Phase Groups	160
Running Weighted Average Simultaneous Switching Output (WASSO)	
Analysis	161
Running WASSO Analysis (Spartan-3, Virtex-4, Virtex-5)	161
Viewing the WASSO Analysis Results	162
Exporting Package Pin Information	163
Exporting an I/O Port List	164

Chapter 6: Creating and Analyzing the RTL Design

Using the Project Environment	165
Using the Sources View	165
Viewing Source File Properties	168
Adding Sources to the Project	168
Adding Source Files or Directories	168
Creating a New Source File	170
Updating RTL Source Files	170
Using the RTL Editor	171
Using the RTL Editor Specific Popup Menu Commands	172
Using the Find in Files Command to Search Source Files	172
Elaborating and Analyzing the RTL Design	173
Using the RTL Netlist View	175
Using the RTL Hierarchy View	176
Viewing the Resource Estimates for Modules	176
Analyzing the RTL Schematic	177
Searching for Objects Using the Find Command	178
Running RTL DRCs	178

Chapter 7: Implementing a Design

Running Synthesis	185
Creating and Launching Synthesis Runs	185
Synthesis Methodology Tips	190
Running Implementation	191
Creating and Launching Implementation Runs	191
Implementing Pblocks	194
Monitoring and Configuring Runs	196
Using the Design Runs View	196
Viewing and Modifying Run Properties	197
Managing Runs	202
Launching Existing Runs	202
Resetting Runs	204
Deleting Runs	204
Importing Run Results	205
Importing Synthesis Run Results	205
Importing Implementation Run Results	207
Running Bitgen on an Implementation Run	208
Creating Strategies	209
Creating Synthesis and Implementation Strategies	209
Creating Common Group Strategies	211
Executing Runs on Multiple Linux Hosts	211
Limitations	211
Configuring Remote Hosts (Linux Only)	212
Interfacing with ISE Outside of PlanAhead	214
Exporting Constraints	214
Exporting Netlist	214
Exporting Pblocks for ISE Implementation	215
Importing ISE Implementation Results	216

Chapter 8: Analyzing the Design

Using the Floorplanning Environment	221
Using the Device View	221
Using the Schematic View	227
Using the Instance Hierarchy View	236
Using the Properties View	238
Using the Netlist View	239
Using the Constraints View	243
Using the Physical Hierarchy View	247
Using Common Popup Menu Commands	251
Analyzing the I/O Pinout and Clock Logic	252
Analyzing the RTL Design	252
Analyzing the Synthesized Design	253
Reporting Design Resource and Device Utilization Statistics	253
Exploring the Logical Hierarchy	258
Analyzing the Hierarchical Connectivity	258
Running Design Rule Checks (DRC)	259
Running Timing Analysis	264
Analyzing Timing Results	266
Searching for Objects Using the Find Command	270

Chapter 9: Analyzing Implementation Results

Importing ISE Placement and Timing Results	273
Importing ISE Placement and Timing Results With New Project Wizard	273
Importing ISE Placement and Timing Results from PlanAhead Runs	273
Importing ISE Placement and Timing Results from Outside PlanAhead	273
Importing ISE Placement and Timing Results from Project Navigator	273
Analyzing Placement and Timing Results	274
Exploring Xilinx TRCE Results	274
Viewing Timing Paths in the Device View	274
Viewing Timing Paths in Schematic View	275
Exploring Logic Connectivity	276
Using the Show Connectivity Command	276
Expanding Logic in the Schematic View	277
Tracing Logic Paths using the Properties Dialog Box Connectivity Tabs	278
Highlighting Selected Objects	280
Highlighting Objects	280
Unhighlighting Objects	280
Highlighting Placed Modules	280
Using the Select Primitives and Highlight Primitives Commands	280
Marking Selected Objects	281
Marking Objects	281
Removing Marks	282
Displaying Design Metrics	282
Using the Metrics View	282
Displaying the Metric Maps in the Device View	283
Hiding Metric Map Display	284
Using the Metrics Results View	284
Configuring the Metric Ranges	285

Chapter 10: Floorplanning the Design

Floorplanning Overview	287
Floorplanning Methodology Tips	287
Partitioning the Design by Creating Pblocks	288
Using the Draw Pblock Command	288
Using the New Pblock Command	290
Creating Multiple Pblocks with the Create Pblocks Command	290
Creating Non-Rectangular Pblocks	292
Creating Nested Child Pblocks	293
Creating Clock Region Pblocks	294
Working with Pblocks	296
Understanding Pblock Graphics	296
Assigning Logic to Pblocks	298
Unassigning Logic from Pblocks	299
Moving a Pblock	299
Stretching a Pblock	300
Using the Set Pblock Size Command	300
Modifying Non-Rectangular Pblocks	301
Removing a Pblock Rectangle	301
Renaming a Pblock	302
Deleting a Pblock	302

Viewing or Changing Pblock Properties	302
Setting Pblock Logic Types Ranges	305
Setting Attributes for Pblocks	306
Using Resource Utilization Statistics to Size Pblocks.	307
Placing Pblocks Based on Connectivity	309
Displaying Bundle Net Properties	310
Adjusting Bundle Net Defaults	311
Using the Automatic Pblock Commands	311
Automatically Creating Pblocks	311
Running the Automatic Pblock Placer	313
Working with Placement LOC Constraints	315
Understanding Fixed and Unfixed Placement Constraints	315
Understanding Site and BEL Level Constraints	315
Assigning Site Location Placement Constraints (LOCs)	316
Assigning BEL Placement Constraints (BELs)	316
Adjusting the Visibility of Placement Constraints	317
Viewing Logic Connectivity using Show Connectivity Mode	318
Moving Placement Constraints	318
Deleting Selected Placement Constraints	319
Selectively Clearing Placement Constraints	319
Moving Pblocks with Placement Constraints Assigned	324
Locking Placement During ISE Implementation	324
Setting Prohibits	324
Using IP Reuse Capabilities	325
IP Reuse Overview	325
Creating and Reusing an IP Module	325
Duplicating Placement for Identical Modules	328

Chapter 11: Debugging the Design with ChipScope

Overview of ChipScope Integration in PlanAhead	329
Requirements and Limitations When Using Core Insertion Flow	330
Using the Core Insertion Flow	330
Deciding Which Debug Core Insertion Mode to Use	331
Selecting Nets for Debug	331
ChipScope Wizard-based Debug Core Insertion	332
Using the ChipScope Window to Add and Customize Debug Cores	334
Implementing the Design	340
Invoking the ChipScope Analyzer	340

Appendix A: Menu and Toolbar Commands

Main Menu Commands	341
File Menu	341
Edit Menu	342
View Menu	343
Tools Menu	343
Window Menu	344
Select Menu	345
Layout Menu	346
Help Menu	347

Toolbar Commands	347
Appendix B: Installing Releases with XilinxUpdate	
PlanAhead Release Strategy	351
Running XilinxUpdate	351
Automatically Checking for Updates	352
Appendix C: Configuring SSH Without Password Prompting	
Setting Up SSH Key Agent Forward	355
Index	357

Understanding the PlanAhead Design Flow

This chapter contains the following sections:

- [“PlanAhead Design Flows”](#)
- [“Input and Output Files”](#)
- [“PlanAhead Terminology”](#)

PlanAhead Design Flows

The PlanAhead™ software can be used in various ways at different points in the FPGA design flow. It can be used as a complete flow management tool from RTL development through bitstream generation, or for I/O pin planning, RTL netlist analysis, implementation result design analysis, floorplanning, or ChipScope™ tool core insertion and debugging.

Through analysis and floorplanning, physical constraints are applied to help control the implementation of the design. The PlanAhead environment enables exploration and experimentation with various implementation strategies. All of the implementation attempts and data are completely managed from within the PlanAhead environment. PlanAhead is also used after implementation to analyze the ISE® software placement and timing results in order to improve the performance of the design. Additional physical constraints derived from the imported results may also be used to lock placement during subsequent implementation attempts. The flow chart shown below illustrates the common design flows as well as the various input and output formats of PlanAhead.

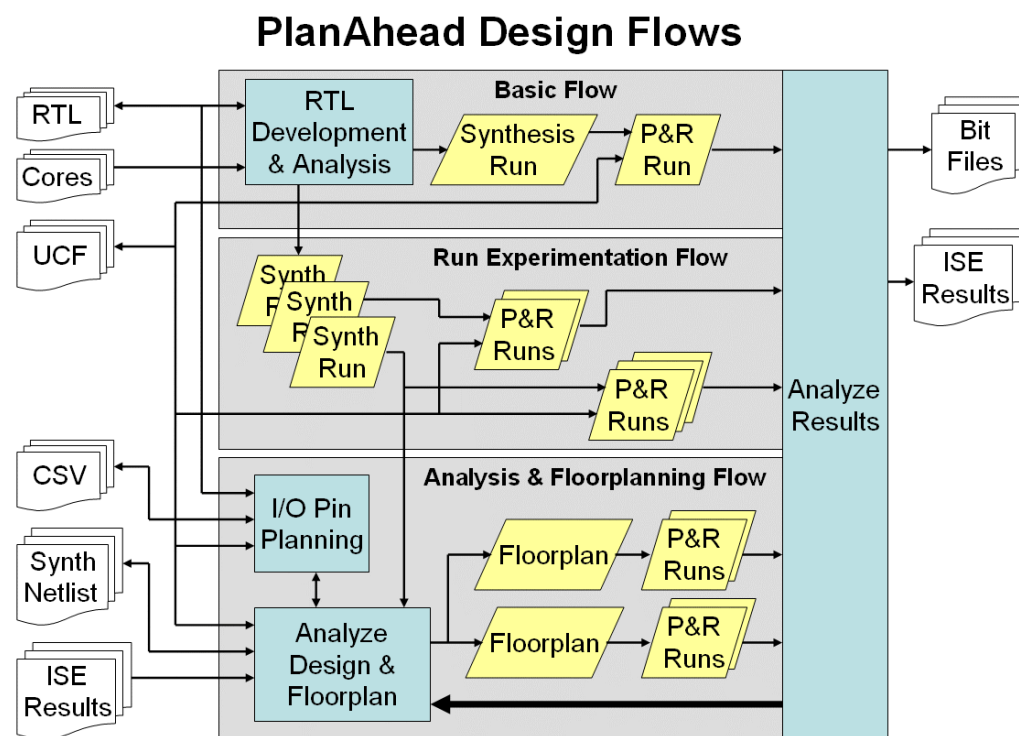


Figure 1-1: PlanAhead Design Flows with Inputs and Outputs

The addition of a logic synthesis environment and front to back process is one of the biggest additions to PlanAhead in the 11 release. You can now import RTL sources, synthesize the logic, implement the synthesized netlist, analyze the implementation results, floorplan and experiment with implementation options and generate bitstreams. It is going a long way toward providing a comprehensive front to back design environment and solution.

PlanAhead manages the design flow and data to accommodate experimentation with multiple synthesis and implementation runs simultaneously. Previously, PlanAhead projects consisted of a single imported netlist. You could create multiple floorplans for experimentation, which were basically constraint sets that all leveraged that single version of the netlist. Creation of a floorplan was always required and the netlist was always loaded into memory during a PlanAhead session.

Basic Design Flow

The basic flow enables users to easily import RTL sources, synthesize and implement the design and view results. It is often used to implement the design initially. If further design analysis and floorplanning is desired, the design analysis flow should be used.

Run Experimentation Flow

You can experiment with multiple synthesis or implementation runs using different strategies. A set of proven PlanAhead strategies is shipped with the tool or you can define your own.

The PlanAhead synthesis and implementation environment enables you to configure, launch, and monitor multiple synthesis runs using the Xilinx® XST synthesis tool. You can define reusable *strategies* for synthesis runs. As an example, you might create strategies for power, performance, or area optimization. You'd then assign these strategies to individual runs, and launch them simultaneously or serially. The synthesis run results are displayed interactively and report files are accessible.

Running multiple synthesis strategies results in multiple netlists being created, and stored within the PlanAhead project. PlanAhead then enables you to interactively load the various versions of the netlist into the environment for analysis. During the netlist import, Floorplans can be created for I/O pin planning, device analysis, floorplanning and implementation.

Implementation runs can be created from any completed PlanAhead synthesis run, floorplan, or imported third-party synthesized netlist. Creating floorplans allows you to experiment with various logical constraints, physical constraints, or alternate devices. You can define reusable strategies for implementation runs. As an example, you might create strategies for the various map logic optimization options or par effort levels. You'd then assign these strategies to individual runs and launch them simultaneously or serially. The implementation run results are displayed interactively and the command report files are accessible.

Design Analysis and Floorplanning Flow

PlanAhead has extensive design analysis and floorplanning capabilities that can be utilized at various stages of the design process. RTL can be elaborated and analyzed prior to synthesis. Synthesized netlists can be explored using target devices and constraints. Implementation runs can be imported into the PlanAhead environment for further analysis and floorplanning. Implementation results derived from using command line tools can also be imported.

Specific flow features for design analysis and floorplanning are covered in various sections of this document.

Input and Output Files

Inputs to PlanAhead

This section briefly describes the formats and processes used while importing design data.

The input files are as follows:

- “RTL Source Files (Verilog, VHDL, or other design text files)”
- “Xilinx Cores (NGC / NGO)”
- “XST Constraint Files (XCF)”
- “I/O Port Lists (CSV)”
- “I/O Port Lists (HDL - Verilog or VHDL)”
- “Top-Level Netlists (EDIF)”
- “Module-Level Netlists (EDIF)”
- “Top-Level Netlists (NGC)”
- “Constraint Files (UCF / NCF)”

- “Xilinx ISE Placement Results (NCD / XDL)”
- “Xilinx TRCE Timing Results (TWX/TWR)”

While reading the input files, PlanAhead writes out any errors, warnings and messages in to the `planAhead.log` file. These messages are also displayed in the PlanAhead Console view.

RTL Source Files (Verilog, VHDL, or other design text files)

Verilog and/or VHDL files can be imported and elaborated to analyze the logic or modify the source. The original source files can be referenced and left in place or they can be copied into the project for portability. The search directories are specified when importing RTL source files. All recognized files and file types contained in the directories will be imported into the Project.

Xilinx Cores (NGC / NGO)

PlanAhead can support designs which use NGC format netlists, such as Xilinx core files, and XST output netlists. The software behaves differently depending on whether the NGC file is a top-level netlist or a module-level netlist. Upon netlist import, PlanAhead will automatically convert NGC and NGO format core files to EDIF. NGC format files get converted using the Xilinx **ngc2edif** command. NGO format files get converted to EDIF using the Xilinx **ngcbuild** and **ngc2edif** commands.

For secure cores, the LUT equations are stripped out of the converted EDIF. These types of NGC or NGO core files have usually been meticulously “hand placed” for maximum performance. Chances are that floorplanning will not improve their performance. It is not recommended that you floorplan logic inside the core modules. However, you can floorplan the location for the entire core and all surrounding logic very effectively.

During the Export Floorplan and Save Floorplan commands (described in [Chapter 10, “Floorplanning the Design”](#)), the NGC and NGO core logic is filtered out of the netlist and “black-box” modules are created. The original source NGC and NGO core files are copied into the save or export directory. This ensures that these original NGC and NGO core files are used during ISE implementation.

Note: The output log for the **ngc2edif** command can be viewed in the PlanAhead terminal window where PlanAhead was invoked. Occasionally, the **ngc2edif** command produces EDIF that is unusable or that now has discrepancies with the accompanying NCF constraints. Please report these issues to Xilinx along with the data to reproduce it. You may continue to floorplan without the cores imported. PlanAhead creates black boxes for the missing logic. Then, you’ll need to copy the NGC core files netlists to the ISE run directory.

XST Constraint Files (XCF)

PlanAhead supports adding XST Constraints Files (XCF) format as sources to configure XST synthesis runs.

I/O Port Lists (CSV)

A Comma Separated Values “CSV” format file can be imported to populate the I/O Ports view within PinAhead. You can then assign these I/O Ports to physical package pins to define the device pin configuration. Refer to [Chapter 5, “I/O Pin Planning”](#) for more information about the CSV file content and format.

I/O Port Lists (HDL - Verilog or VHDL)

A Verilog or VHDL format file header that contains ports can be imported to populate the I/O Ports view within PlanAhead. You can then assign these I/O Ports to physical package pins to define the device pin configuration. Refer to [Chapter 5, “I/O Pin Planning”](#) for more information about the HDL file content and format.

Top-Level Netlists (EDIF)

Currently, PlanAhead supports the importing of EDIF netlists. The netlist should be synthesized for a Virtex[®]-4, Virtex-5, Virtex-6, Spartan[®]-3 or Spartan-6 device.

PlanAhead can construct the design using multiple EDIF netlists supporting a hierarchical design methodology. When you select the top-level logic, lower-level modules are imported automatically. Incremental netlist import capabilities allow netlist updates at any level of design hierarchy. In-process floorplanning constraints are maintained through iterations.

Module-Level Netlists (EDIF)

PlanAhead can construct the design using multiple EDIF netlists supporting a hierarchical design methodology. When you select the top-level logic, lower-level modules are imported automatically. You can define a search path to locate the design modules. The advantage to this process is more flexibility when updating the design. PlanAhead has a robust incremental netlist import capability allowing netlist updates at any level of the design hierarchy.

Top-Level Netlists (NGC)

If top-level NGC files are used, PlanAhead automatically converts them to EDIF format during the netlist import process. Top-level NGC format files generated with XST get converted using the **ngc2edif** command. When exporting the design from PlanAhead for ISE implementation, the netlist for the top-level logic is exported in EDIF format to be used during **ngdbuild**. This is different than the export process for core-level NGC or NGO files. See below for more information about core or module-level netlists.

Constraint Files (UCF / NCF)

PlanAhead supports importing UCF and NCF format files for timing and physical constraints. PlanAhead can import multiple UCF files allowing separation of physical constraints, I/Os and timing constraints.

Module-level constraints that are specific to cores, such as NCF files, may also be imported. For more information, see [“Importing Module-Level Constraints.”](#)

PlanAhead supports all of the UCF constraints supported by Xilinx. Refer to the Xilinx *Constraints Guide* for more detailed information about UCF constraints and supported syntax.

Xilinx ISE Placement Results (NCD / XDL)

PlanAhead can import ISE placement results using XDL format data. XDL data is created automatically when implementation runs are launched from PlanAhead.

Once the ISE commands have completed satisfactorily, an XDL format file can be created from the `<placed_design_name>.ncd` file. XDL files can be created and placement can be imported for individual blocks or for the entire design.

When using the Import Placement command, PlanAhead will automatically run the XDL command when a `<placed_design_name>.ncd` file is selected in the Import Placement dialog box.

To run the command by hand, the file syntax is detailed below:

```
xdl -ncd2xdl <placed_design_name>.ncd
```

Running this command will result in creation of a `<placed_design_name>.xdl` file.

The XDL command status is displayed in the PlanAhead Terminal window.

Xilinx TRCE Timing Results (TWX/TWR)

PlanAhead can import the timing report generated by the Xilinx **trce** command. This includes TWX and TWR files. Once imported, all signal tracing and selection is available through the TimeAhead interface.

Outputs for Reports

This section briefly describes the files created during normal PlanAhead design operations. These files can contain valuable information and are described here. The last two reports are not automatically generated, and require user interaction to create.

The output log and report files are as follows:

- “I/O Pin Assignment (CSV)”
- “I/O Pin Assignment (RTL - Verilog or VHDL)”
- “Log File (planAhead.log)”
- “Journal File (planAhead.jou)”
- “Error Log Files (planAhead_pidxxxx.debug & hs_err_pidxxxx.log)”
- “DRC Results (results_x_drc.txt)”
- “TimeAhead Results (Excel file)”
- “Netlist Module, Pblock, and Clock Region Statistics Reports”
- “SSN Analysis Report”
- “WASSO Analysis Reports”

I/O Pin Assignment (CSV)

This comma separated value (CSV) format file contains all of the I/O port assignment and relative package pin information. This file is intended to be used for RTL port header definition and PCB schematic symbol generation. Refer to [Chapter 5, “I/O Pin Planning”](#) for more information.

I/O Pin Assignment (RTL - Verilog or VHDL)

This Verilog or VHDL format file contains all of the I/O port assignments defined as ports in the file header in a legal language format. This file is intended to be used for RTL port header definition. Refer to [Chapter 5, “I/O Pin Planning”](#) for more information.

Log File (planAhead.log)

The log file, `planAhead.log`, captures the contents of the messages created from running PlanAhead commands. The file is created in the PlanAhead invocation directory (for

Linux), and in `C:\Documents and Settings\<user>\Application Data\HDI` (for Windows). It can be displayed in PlanAhead by selecting **Window > View Log File**.

Journal File (planAhead.jou)

The journal file, `planAhead.jou`, captures all of the TCL commands from PlanAhead session that was invoked. The file is created in the PlanAhead invocation directory (for Linux), and in `C:\Documents and Settings\<user>\Application Data\HDI` (for Windows). The journal file can be replayed to reproduce the session's previous commands. TCL scripts can be created by copying commands from the journal file for replay later in PlanAhead. It may be necessary to edit this file to remove any erroneous commands or commands from multiple PlanAhead sessions prior to replay.

Note: Not every action in PlanAhead will log a TCL command into the journal file.

Error Log Files (planAhead_pidxxxx.debug & hs_err_pidxxxx.log)

The error files can provide valuable information for debugging PlanAhead crashes. If PlanAhead issues a dialog box that warns of an internal exception error, the error files are stored in the PlanAhead invocation directory (on Linux), and in `C:\Documents and Settings\<user>\Application Data\HDI` (on Windows). When you open a case with Xilinx Technical Support, include any generated error log files, the PlanAhead journal file (`planAhead.jou`), and the log file (`planAhead.log`). These files contain no design data.

DRC Results (results_x_drc.txt)

The results from the Design Rule Check (DRC) are reported in the `results_x_drc.txt` files created in the PlanAhead invocation directory (on Linux), and in `C:\Documents and Settings\<user>\Application Data\HDI` (on Windows). Each time DRC is run, a new file is produced in the PlanAhead Project directory with a corresponding number to results listed in the PlanAhead DRC dialog box.

TimeAhead Results (Excel file)

The results from TimeAhead timing analysis can be exported into a text file. To export the data, select the Export to Text File icon in the Timing Results view.

Netlist Module, Pblock, and Clock Region Statistics Reports

The resource statistics displayed in the Instance Properties, Clock Region and Pblock Properties View can be exported to an Microsoft Excel format file. This information includes resource utilization, RPM and carry chain sizes, clocks and clocked instances, and other relevant resource data.

To export the data, select the Save statistics to file icon from the Statistics tab of the Instance, Clock Region or Pblock Properties View. The dialog box allows you to define the information to include in the report as well as how many levels of hierarchy to report. A browser will allow you to specify a file name and location.

SSN Analysis Report

The results from the PlanAhead Simultaneous Switching Noise (SSN) analysis can be exported to a CSV report file by specifying a file name and location in the *Run SSN Analysis* dialog box.

WASSO Analysis Reports

The results from the PlanAhead Weighted Average Simultaneous Switching Output (WASSO) analysis can be exported to a text report file by specifying a file name and location in the *Run WASSO Analysis* dialog box.

Outputs for Environment Defaults

This section briefly describes the files created during normal PlanAhead design operations. These files can contain valuable information and are described here. The last two reports are not automatically generated, and require user interaction to create.

The output files are as follows:

- “View Display Options File (planAhead.ini & <theme_names>.patheme)”
- “Window Layout Files (<layoutname>.layout)”
- “Shortcut Schema (default.xml)”
- “Strategy Files (<strategyname>.psg)”

View Display Options File (planAhead.ini & <theme_names>.patheme)

The initialization file, `planAhead.ini`, captures all of the current Tool > Options settings which include display color and other viewing options for the entire PlanAhead Environment. Upon exiting PlanAhead, custom user settings are saved to this file for future PlanAhead sessions. The file is automatically created in your home directory.

On Windows, this is often in ‘C:\Documents and Settings\<Username>\Application Data\HDI\<version_number>\planAhead.ini’. When PlanAhead is invoked, the file is automatically imported from the PlanAhead installation directory first, and then from the ‘C:\Documents and Settings\Owner\Application Data\HDI\<version_number>’ directory, if it exists.

On Linux, this is often in ‘~/ .HDI/planAhead.ini’. When PlanAhead is invoked, the file is automatically imported from the PlanAhead installation directory and then from the ‘~/ .HDI’ directory, if it exists.

You can save custom Theme files for use in future PlanAhead sessions by clicking the **Save As** button in the Themes dialog box (accessed using **Tools > Options > Themes**). A pull-down selection menu allows you to select the desired Theme file to use during the active PlanAhead session. For more information, see “[Creating and Using a Customized Theme](#).”

When you select the **PlanAhead Light/Dark Theme** buttons of the View Options dialog box, the ‘C:\Documents and Settings\Owner\Application Data\HDI\<version_number>\planAhead.ini’ file (on Windows) and the ‘~/ .HDI/planAhead.ini’ file (on Linux) is overridden with these preset defaults. To avoid loss of any custom settings, keep a backup of the custom settings file.

Window Layout Files (<layoutname>.layout)

Window layout files are created using the Save Layout As or Save as Default Layout commands and will save the current PlanAhead Desktop view configuration to be recalled at a later time. The configurations for both the Floorplan and Project viewing environments are stored in the following subdirectories.

On Windows, this is often in ‘C:\Documents and Settings\<Username>\Application

Data\HDI\<version_number>\layouts\floorplan_layout or project_layout'. The overall PlanAhead window size and location is saved in the 'C:\Documents and Settings\<Username>\Application Data\HDI\<version_number>\layouts\application_layout directory.

On Linux, the data is created and stored in ~/ .HDI directory.

Shortcut Schema (default.xml)

Accelerator key definitions, or "shortcut schema," are created when using the Options dialog box. These schema define a mapping from keyboard shortcuts to PlanAhead commands. For example, the keystroke **Ctrl + F** by default maps to the **Edit > Find** command. You may define and configure multiple schemas, all of which are stored in the default.xml file.

On Windows, this is often in C:\Documents and Settings\<Username>\Application Data\HDI\<version_number>\shortcuts.

On Linux, the data is created and stored in ~/ .HDI/shortcuts.

Strategy Files (<strategyname>.psg)

Strategy files contain your specified default command line options for all of the ISE implementation commands. You can apply a strategy to any given ISE attempt using PlanAhead. You can either create strategies from scratch or copy one of the factory supplied strategies. User defined strategies are stored in your home directory.

On Windows, this is often in 'C:\Documents and Settings\<Username>\Application Data\HDI\<version_number>\strategies'.

For Linux, the data is created and stored in ~/ .HDI/strategies.

Outputs for Project Data

This section briefly describes the files created for saved PlanAhead Projects and Floorplans. These files are maintained by PlanAhead and should not be modified manually. The project output files are as follows:

- "Project Directory (<projectname>)"
- "Project File (<projectname>.ppr)"
- "Project Data Directory (<projectname>.data)"
- "Project Data - Netlist Subdirectory (netlist)"
- "Project Data - Floorplan Subdirectories and Files (<floorplan_name>)"
- "Project RTL Directory (<projectname>.srcs)"

Project Directory (<projectname>)

When a new Project is created, PlanAhead creates a Project directory to store the Project File, the Project data directory and the ISE implementation results. The Project directory has the same name as the Project name entered in the New Project Wizard.

Project File (<projectname>.ppr)

The Project PPR file stores the state of the Project. It contains information about the netlist and the various Floorplans contained in the Project. The file is continuously maintained while PlanAhead is invoked. It does not require saving. This file should not be edited manually.

The PPR file is the item selected in the PlanAhead browser when opening an existing Project.

Project Data Directory (<projectname>.data)

The Project Data Directory stores all of the Floorplan and netlist-related data contained in the Project. These folders are maintained by PlanAhead and do not require your attention.

Caution! Modifying any of these files could result in Project data corruption.

Project Data - Netlist Subdirectory (netlist)

A subdirectory called `netlist` contains a copy of the netlist files for the entire design.

For RTL based Projects, a subdirectory for each synthesis run is created containing the netlist produced. It is refreshed each time the synthesis run is reset.

For netlist based Projects, a single netlist directory is created containing the imported netlist, including copies of all NGC core files used in the design. The **File > Update Netlist** command can be used to update the contents of this subdirectory

Project Data - Floorplan Subdirectories and Files (<floorplan_name>)

As Floorplans are created, matching subdirectories are created under the `<projectname>.data` directory.

The files found in the Floorplan directory are listed below.

- `*.ucf` – All imported UCF file names (may differ from input files)
- `fp.ucf` – Contains current PlanAhead constraints for the Floorplan
- `iseloc.xml` – Used to differentiate the PlanAhead fixed placement constraints from the unfixed placement constraints imported from ISE
- `pfi.xml` – Contains target device for Floorplan
- `pfp.xml` – Contains current PlanAhead experiment information for the Floorplan
- `expX` subdirectories – Contains PlanAhead experiment information about each run

Project RTL Directory (<projectname>.srcs)

The Project sources directory stores all of the HDL Source files imported into the Project. These folders are maintained by PlanAhead and do not require your attention.

Caution! Modifying any of these files could result in Project data corruption.

Outputs for ISE Implementation

This section briefly describes the files created during PlanAhead ISE implementation design operations. These files are maintained by PlanAhead and should not be modified manually.

The output files for ISE implementation are as follows:

- “Run Directory (<projectname>.runs)”
- “EDIF Netlists (.edf)”
- “Xilinx Cores (.ngc/.ngo)”
- “ChipScope Core Netlists (.ngc)”
- “Constraint Files (.ucf)”
- “ISE Launch Scripts (jobx.bat/sh & runme.bat/sh & .<ISE_command>.rst)”

Run Directory (<projectname>.runs)

PlanAhead allows you to launch and queue multiple ISE implementation attempts or “Runs”. You are prompted to enter a location to create the Runs directory. The default location is in the Project directory.

Each run directory will contain a complete EDIF netlist and UCF constraint file for the run. A run script to launch the ISE commands with your specified options is also created in each PlanAhead Run directory.

Each Run directory contains all implementation design data including the netlist and the constraints files. When a satisfactory implementation result is achieved, the entire Run directory can be easily copied and archived because it is self contained.

EDIF Netlists (.edf)

Currently, PlanAhead exports EDIF format ASCII netlist files. These files are created during the following commands:

- **Run Implementation and Launch Runs (PlanAhead)**
- **File > Export Netlist**
- **File > Export Pblocks**
- **File > Export IP**

Run Implementation and Launch Runs

The purpose of launching PlanAhead Runs is to automatically export the files required to implement the PlanAhead Runs and to launch the ISE commands with the options specified in the Strategy applied to the Run.

EDIF and UCF data is exported automatically when the Launch Runs command is used. When a Run is launched, a run directory is created containing a single EDIF format netlist file and UCF format constraint file for the entire top-level design. The file names correspond to the original top-level netlist name contained in the originally imported EDIF file.

If NGC/NGO format module netlist files are used, they are copied to each Run directory.

The PlanAhead General Run Properties indicate where the actual Run directory resides on disk.

Exported Netlists

The purpose of exporting a Netlist is to supply the design EDIF file for ISE implementation outside of the PlanAhead environment. When a Netlist is exported, the original logical netlist hierarchy is maintained in the output netlist. You can specify an output file name in the Export Netlist dialog box.

Exported Pblocks

The purpose of exporting a Pblock is to write out the EDIF and UCF files for the specified Pblocks to use for ISE implementation outside of the PlanAhead environment.

When a Pblock is exported, PlanAhead will create the netlist based on the Pblock logic assignments. The resulting EDIF and module port list are derived and use the PlanAhead physical hierarchy structure. A single EDIF netlist is created and it includes all of the logic assigned to the Pblock. This provides ultimate flexibility when using a block-based implementation strategy.

The exported Pblock files consist of a single netlist file and constraint file for each of the selected Pblocks during export. A block-level directory structure is automatically created and maintained simplifying a block-based ISE approach. Exporting selected Pblocks will create `<pblockname>_CV` subdirectories containing `<pblockname>_CV.edn` and `<pblockname>_CV.ucf` files.

Exported IP

The purpose of exporting IP is to write out the EDIF and UCF files for specified Netlist modules to be used for creating reusable IP blocks.

The Export IP command is run on a selected netlist hierarchy in the design. A RPM can also be produced as output for the IP module. The exported files will include the EDIF netlist and UCF physical constraints written in the original logical netlist format. This allows easier implementation in the next design by keeping the interface identical. The exported EDIF file can be used to populate any number of “black-boxed” RTL modules in the new design. If using XST, the exported EDIF file can be used to derive some timing data. The exported UCF file can be used to recreate the Pblock placement constraints. Identical placement can be duplicated for multiple modules by moving the modules after they are imported.

Xilinx Cores (.ngc/.ngo)

During the Export Floorplan and Save Floorplan commands (described in [Chapter 10, “Floorplanning the Design”](#)), the NGC and NGO core logic is filtered out of the netlist and “black-box” modules are created. The original source NGC and NGO core files are copied into the save or export directory. This ensures that these original NGC and NGO core files are used during ISE implementation.

Note: The output log for the `ngc2edif` command can be viewed in the PlanAhead terminal window where PlanAhead was invoked. Occasionally, the `ngc2edif` command produces EDIF that is unusable or that now has discrepancies with the accompanying NCF constraints. Please report these issues to Xilinx along with the data to reproduce it. You may continue to floorplan without the cores imported. PlanAhead creates black boxes for the missing logic. Then, you’ll need to copy the NGC core files netlists to the ISE run directory.

ChipScope Core Netlists (.ngc)

PlanAhead is integrated with ChipScope enabling the ILA cores to be inserted and configured interactively. An NGC format netlist for the core is compiled when the core is implemented. It is placed in the Project netlist directory and copied to each implementation run directory as runs are launched. Refer to the [Chapter 11, “Debugging the Design with ChipScope”](#) for more information.

Constraint Files (.ucf)

Currently, PlanAhead writes UCF format ASCII files containing timing and physical constraints that are used for ISE. These files are created during the following commands:

- **Run Implementation and Launch Runs (PlanAhead)**
- **File > Export Constraints**
- **File > Export Pblocks**
- **File > Export IP**

Run Implementation and Launch PlanAhead Runs

The exporting of EDIF and UCF data is done automatically when the Launch Runs command is used. When a Run is launched, a run directory is created containing the original logic hierarchy in the output netlist. The exported files for the run consist of a single EDIF format netlist file and a UCF format constraint file for the entire top-level design. The file names correspond to the original top-level netlist name of the imported EDIF file.

Exported Constraints

When constraints are exported, preservation of the original UCF file content and structure is attempted, including comments.

You can specify the output constraints file in the Export Constraints dialog box.

Exported Pblocks

When a Pblock is exported, PlanAhead will derive the netlist hierarchy based on the Pblock assignments. The resulting UCF will reference the PlanAhead physical hierarchy structure to match the exported EDIF netlist names. This provides ultimate flexibility when using a block-based implementation strategy.

The exported Pblock files consist of a single netlist file and constraint file. A block-level directory structure is automatically created and maintained simplifying a block-based ISE approach. Exporting selected Pblocks will create *<pblockname>_CV* subdirectories containing *<pblockname>_CV.edn* and *<pblockname>_CV.ucf* files.

Exported IP

The Export IP command is run on a selected module instance in the design and will export the Pblock logic and placement constraints. The exported files will include the EDIF netlist and UCF physical constraints in the original logical netlist format. This allows for easier implementation in the next design by keeping the interface identical. The exported UCF file can be used to recreate the Pblock placement constraints. Identical placement can be duplicated for multiple modules by moving the modules after they are imported.

ISE Launch Scripts (jobx.bat/sh & runme.bat/sh & .<ISE_command>.rst)

ISE launch scripts are automatically created when using the PlanAhead Run Design commands. These scripts contain commands and command-line options specified in the PlanAhead Strategy. The jobx.bat/sh scripts are located under the Project Runs directory in a .jobs subdirectory and will sequentially launch each Run selected. The script will call each Run specific runme.bat | sh script. These scripts can also be launched independently.

The `.<ISE_command>.rst` files are created in the run directory in order for PlanAhead to track the progress and status of runs. PlanAhead will read these files upon opening the Project and will display the status of the run.

PlanAhead Terminology

Project

Each PlanAhead™ software session initiates an active Project. The Project can be created with various input formats, depending on the design flow being applied.

- RTL source files can be imported to create a Project that is suitable for the RTL through bitstream flow.
- A synthesized netlist can be imported and used in the netlist through bitstream flow.
- An empty Project can be created to explore device resources or to begin I/O pin planning, as described in [Chapter 5, “I/O Pin Planning.”](#)
- There is also a Project creation method which allows results to be imported from a previous command line implementation attempt.

Depending on the Project type created, the Project can contain one or more versions of the netlist. Each with any number of Floorplans along with the implementation attempts or “Runs”. Refer to [“Managing Projects”](#) for more information.

The Project information is stored in directory structure containing a combination of a project file, such as `project_1.ppr`, a Project data directory, such as `project_1.data`, a Project sources directory, such as `project_1.srcs`, and a Project runs directory, such as `project_1.runs`. The sources directory contains all RTL related source files imported into the Project. The data directory contains the netlist directories containing the Project netlists and directories for each Floorplan in the Project. The runs directory contains all ISE implementation attempts created by PlanAhead.

The Project data is maintained automatically by PlanAhead. The tool expects to find project data in the state it left it. Therefore, you should not attempt to modify these files manually.

PlanAhead will automatically restore the state of the Project upon opening it. The Project status including all opened or closed Floorplans and each associated Run are updated and available to you when a Project is re-opened.

Source

Projects can be created with a variety of input file formats. Projects can be created by importing RTL source files in Verilog and VHDL. These files are considered source files. Only Projects created by importing RTL files contain Sources. Projects created by importing synthesized netlists or empty Projects do not contain source files.

Floorplan

Floorplans are merely a set of constraints associated with a selected netlist and device. Creation of Floorplans is not required to use PlanAhead. Implementation runs can be launched using any external User Constraints File (UCF). In order to modify or apply any constraints within PlanAhead, a Floorplan must first be created. Each Floorplan associates an active netlist with the specific placement and timing constraints defined in it. Each Project netlist can support multiple Floorplans using different constraints or devices. Floorplans may be saved, closed, restored, copied, renamed and deleted. Closed

Floorplans remain in the Project but do not open by default upon reopening the Project, which saves system memory.

Netlist

A netlist represents a logical description of the design. A netlist should be hierarchical consisting of a top-level netlist with child netlists for underlying levels of hierarchy ("modules"). PlanAhead RTL-based Projects can contain multiple netlists since multiple synthesis runs are enabled.

Constraint

A constraint can either be a description of the timing of logic, some behavioral requirement, or a physical placement requirement. I/O Port assignments are also defined by constraints.

Physical Block (Pblock)

Design partitions are referred to as physical blocks or Pblocks. Traditionally, a single or group of logic instances are assigned to a Pblock. The Pblock can have an area, such as a rectangle defined on the FPGA device, to constrain the logic. Pblocks can be defined without rectangles, and ISE will attempt to group the logic during placement. Netlist logic placed inside of Pblocks will receive AREA_GROUP constraints for ISE. Pblocks may be specified with specific RANGE types to contain various types of logic only (e.g. SLICE, RAM/MULT, DSP, etc.). Pblocks may also be defined with multiple rectangles to enable non-rectangular shapes to be created, such as 'L' shaped and 'T' shaped.

Instance

Elements in the Netlist referred to as instances include leaf-level logic primitives and hierarchical module components. The module components are referred to as modules in this document.

Module

Elements in the netlist that represent hierarchical module instantiations are referred to as modules or components. Leaf-level primitive logic is referred to as instances or primitives.

Primitive

Elements in the netlist that represent leaf-level logic objects are referred to as primitives (e.g. LUTs, Flip-Flops, etc.).

Run

Each synthesis or implementation attempt is called a Run. Each Run is associated with a specific strategy. You can launch multiple runs either simultaneously with multiple processors or serially. Runs will be queued sequentially with the status displayed in PlanAhead.

Strategy

A Strategy is a predefined set of tool command-line options. You can apply factory delivered Strategies or create your own. Strategies can be applied to individual runs.

Site

PlanAhead displays a tile grid representation of the specific FPGA device resources that can be used to implement the design netlist. Primitive logic sites are displayed and are available for placement of netlist instances. These sites vary in shape and color to differentiate the object types (e.g. RAMs, MULTs, CLBs, DSPs, PPCs, MGTs, etc.). Leaf-level logic can be assigned to specific SLICES with placement constraints “LOCs”, or to gates within the SLICE with LOC and BEL constraints.

Site Placement Constraint (LOC)

Location constraints or LOCs can be assigned to the leaf-level instances that have fixed placement sites assigned to a specific SLICE coordinate. These are different than BEL constraints as they do not lock the logic into specific logic gates within the SLICE. Assigning a LOC constraint will result in a LOC constraint being “fixed” and applied in the exported UCF files for the instance. Depending on zoom level, these LOCs appear in the Device view either as rectangles within their respective assigned sites or logic functions symbols within the site.

BEL Placement Constraint (BEL)

Basic Element (BEL) constraints can be assigned to the leaf-level instances that have placement sites assigned to specific logic device gates. Assigning a BEL constraint will result in a LOC and a BEL constraint being “fixed” and written in the exported UCF files for the instance. Depending on the zoom level, these LOCs appear in the Device view either as rectangles within their respective assigned Sites or as logic functions symbols within the site.

I/O Port

I/O Ports are the user I/Os intended to be assigned to physical package pins. Each I/O signal is defined as a Port

Package Pin

Package Pins are the physical pins of the package to which I/O Ports are assigned. The Package Pins are grouped into I/O Banks. Refer to the device specifications for more information about the Package Pins and I/O Banks.

Creating and Managing Projects

This chapter contains the following sections:

- “Invoking PlanAhead”
- “Using the Getting Started Jump Page”
- “Understanding the Different Types of PlanAhead Projects”
- “Using the Create New Project Wizard to Create a New Project”
- “Managing Projects”
- “Working with Floorplans”

Invoking PlanAhead

Note: Refer to the *Xilinx ISE Design Suite: Installation, Licensing, and Release Notes* for proper installation of this product.

The PlanAhead™ software can be invoked from any directory. However, invoking it from a “project” directory may prove advantageous as project specific log files can be more easily located.

Linux

To invoke PlanAhead type the following command at the Linux command prompt:

```
# planAhead
```

Windows

Double-click on the Xilinx® PlanAhead 11 shortcut icon.



Figure 2-1: Xilinx PlanAhead 11 Icon

The PlanAhead *Start in* folder can be specified by modifying the desktop icon properties to define where the PlanAhead log files will be written.

The PlanAhead Environment will display.

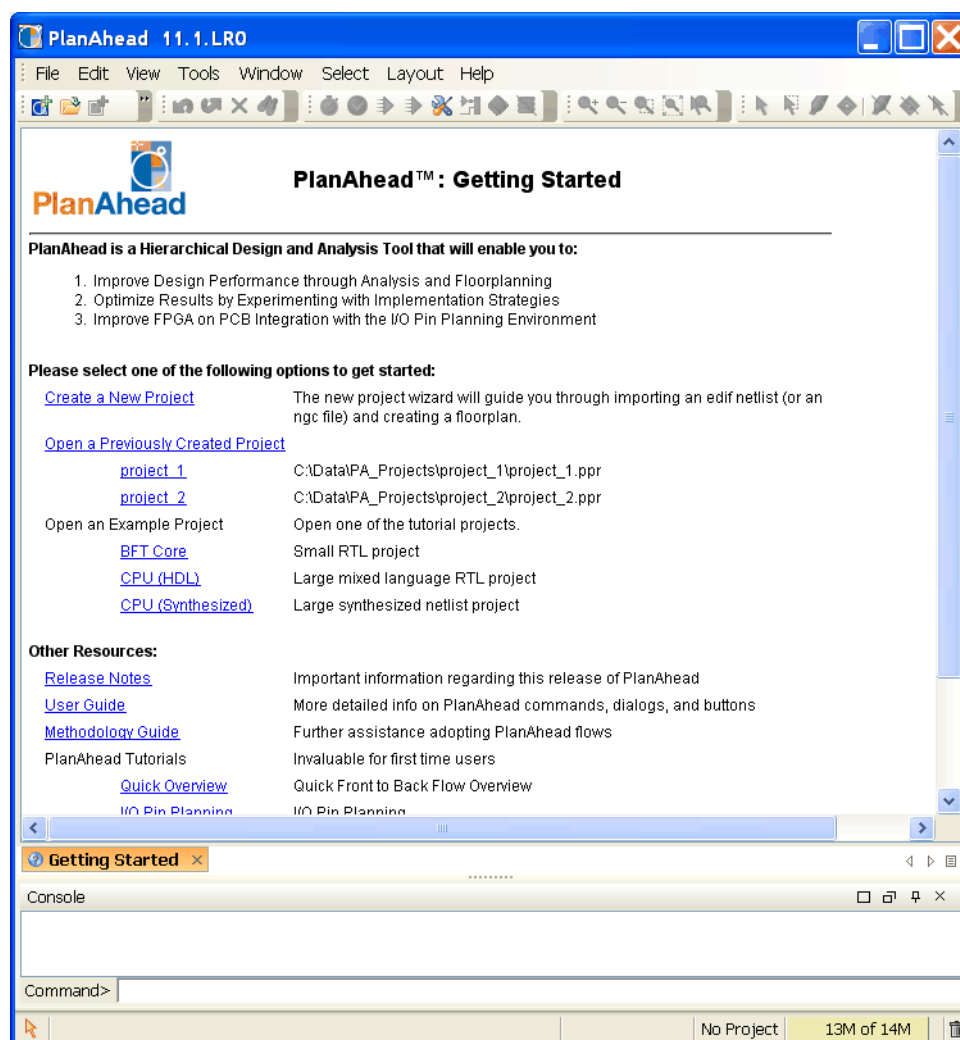


Figure 2-2: The PlanAhead: Getting Started Window

PlanAhead is now available for new or existing Projects to be opened. The PlanAhead Getting Started jump page assists you with creating or opening desired Projects as well as viewing the PlanAhead documentation.

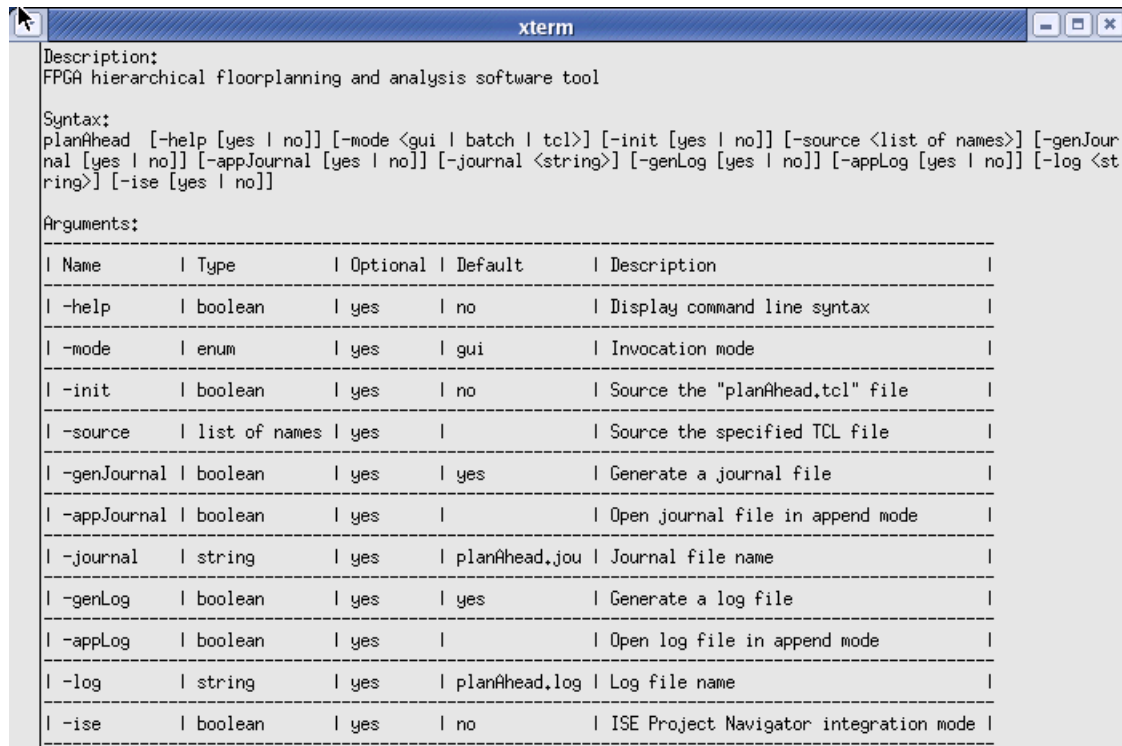
Note: The Getting Started jump page can be displayed by selecting **Help > Getting Started**.

PlanAhead Command Line Options

PlanAhead has several command line options to control the behavior. To view the PlanAhead command line options, type the following command at the command prompt:

```
# planAhead -help
```

A help menu will display in the shell window.



```

Description:
FPGA hierarchical floorplanning and analysis software tool

Syntax:
planAhead [-help [yes | no]] [-mode <gui | batch | tcl>] [-init [yes | no]] [-source <list of names>] [-genJournal [yes | no]] [-appJournal [yes | no]] [-journal <string>] [-genLog [yes | no]] [-appLog [yes | no]] [-log <string>] [-ise [yes | no]]

Arguments:

```

Name	Type	Optional	Default	Description
-help	boolean	yes	no	Display command line syntax
-mode	enum	yes	gui	Invocation mode
-init	boolean	yes	no	Source the "planAhead.tcl" file
-source	list of names	yes		Source the specified TCL file
-genJournal	boolean	yes	yes	Generate a journal file
-appJournal	boolean	yes		Open journal file in append mode
-journal	string	yes	planAhead.jou	Journal file name
-genLog	boolean	yes	yes	Generate a log file
-appLog	boolean	yes		Open log file in append mode
-log	string	yes	planAhead.log	Log file name
-ise	boolean	yes	no	ISE Project Navigator integration mode

Figure 2-3: The *planAhead* Command Arguments

Using a PlanAhead Startup Tcl Script

The PlanAhead **Tools > Run Tcl Script** command can be used to run a startup script. PlanAhead Tcl commands copied from the `planAhead.jou` file can be used to create startup scripts. For more information regarding the PlanAhead journal file, see [“Outputs for Reports”](#).

```

#-----
# PlanAhead version JPA.91.0
# Built by bdeegan on Mon Feb 26 22:27:48 PST 2007
# Start of session at: 2/28/07 11:44:49 AM
# Process ID: 4728
#-----
hdi::project open -file (C:\Data\PlanAhead_Designs\PlanAhead_Tutorial\labs\projects\Project_2\Project_2.ppr)
hdi::project startUpdate -name Project_2 -file (C:\Data\PlanAhead_Designs\PlanAhead_Tutorial\labs\design_files\rev1_top_upd
hdi::project commitUpdate -name Project_2 -all yes
hdi::timing run -name results_1 -project Project_2 -floorplan orig_fp -min_max max -transition rise/fall -sort_by group -int
hdi::drc run -name results_1 -project Project_2 -floorplan orig_fp
hdi::run add -name run_1 -project Project_2 -floorplan orig_fp -flow (ISE 9) -strategy (EA #1)
hdi::run add -name run_2 -project Project_2 -floorplan orig_fp -flow (ISE 9) -strategy (EA #2)
hdi::run schedule -names (run_1 run_2) -project Project_2 -floorplan orig_fp
hdi::run launch -project Project_2 -jobs 1 -scriptsOnly no -allPlacement no -dir (C:\Data\PlanAhead_Designs\PlanAhead_Tutori
#-----
# End of session at: 2/28/07 11:47:57 AM
# Process ID: 4728
#-----

```

Figure 2-4: Example PlanAhead Tcl Script

Using the Getting Started Jump Page

When PlanAhead is invoked, the Getting Started jump page is displayed in the PlanAhead window. You can select the blue underlined command links to invoke specific commands or view documentation, as shown above.

By default, the last 10 previously opened Projects are displayed for re-opening. PlanAhead checks to ensure the Project data is still available before displaying. You can configure how many Projects to list in the General dialog box (available by selecting **Tools > Options > General**).

The PlanAhead documentation is available by selecting the appropriate links to launch a PDF viewer. The documentation is also available in PDF format in the <InstallDir>/doc directory.

Understanding the Different Types of PlanAhead Projects

PlanAhead can be used at different points in the FPGA design flow for various reasons. To accommodate this, different types of PlanAhead Projects can be created. They are differentiated by types of input sources used to create the Project. You can select the type of Project desired during the Create New Project process.

Once a particular Project type is selected, it cannot be migrated to a different type later.

Note: PlanAhead also uses a derivative type of Project to support Partial Reconfiguration designs. This capability is only available to a limited set of customers in the 11 release and is covered in the *Partial Reconfiguration User Guide*.

Empty Projects for I/O Pin Planning

I/O pin planning can be accomplished early in the design cycle by creating an empty Project. I/O ports can be created within PlanAhead or imported with either CSV, UCF or RTL input files. After I/O pin assignment, PlanAhead can create CSV, UCF and RTL output files for use later in the design flow when RTL sources or netlists are available. The output files can also be used to create schematic symbols for use in the printed circuit board (PCB) design.

Empty Projects can also be created to simply explore the logic resources available in the different device architectures.

RTL Source Based Projects

PlanAhead can be used to manage the entire FPGA design flow from RTL creation through bitstream generation. Projects can be created by importing RTL source files as well as precompiled NGC/NGO format Xilinx cores. You can elaborate and analyze the RTL to ensure proper constructs, launch and manage various synthesis and implementation runs, and analyze the design and run results. You can also create Floorplans to experiment with different constraint or device strategies.

Synthesized Netlist Based Projects

You can also create Projects from designs that were synthesized outside of PlanAhead using XST or any supported third-party synthesis tool. PlanAhead will import either EDIF or NGC/NGO format netlists. The netlist can be all inclusive in one file or hierarchical in nature, consisting of multiple module level netlists. You can analyze the logic netlist,

launch and manage various implementation runs, and analyze the design and run results. You can also create Floorplans to experiment with different constraint or device strategies.

Implemented Design Results Based Projects

Projects can also be created to allow analysis of implementation results created outside of PlanAhead using the Xilinx command line tools. The design netlist, implementation, and timing results can be imported to explore timing or placement related issues.

Using the Create New Project Wizard to Create a New Project

The New Project wizard walks through the individual steps to define a Project name and storage location, to import the netlist, and to create an initial Floorplan, including selecting a device and importing the constraints.

To create a new project:

1. Select one of the following commands:
 - ◆ Click the **New Project** toolbar button.



Figure 2-5: New Project Toolbar Icon

- ◆ Click the **Create a New Project** link on the Getting Started jump page.
- ◆ Select **File > New Project**.

The first dialog box of the New Project wizard gives an overview of the wizard intent.

2. To continue, click **Next**.

The Project Name page appears.

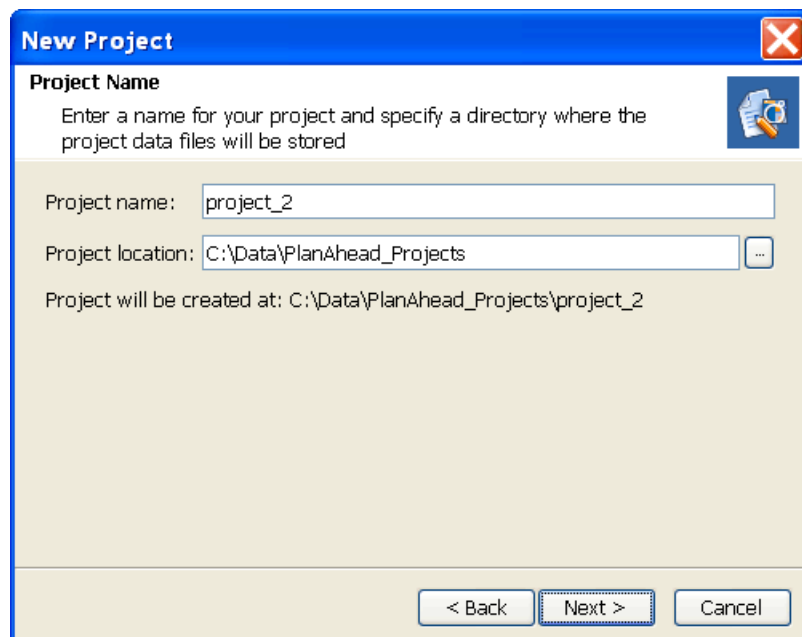


Figure 2-6: New Project Wizard: Project Name Page

Entering a Project Name and Storage Location for the Project

Enter a project name and storage location as follows:

3. In the Project Name page, specify a Project name and disk storage location.
 - ◆ **Project name**—Enter a name to identify the project directory (e.g. project_2).
 - ◆ **Project data files location**—Enter a location to create the project directory.
4. After the above selections are defined, click **Next**.

Selecting the Design Source Data Type

Designate the Design Source input format by selecting either to import RTL sources, or a synthesized (EDIF or NGC) netlist for PlanAhead, or to start with an empty Project for I/O pin planning.

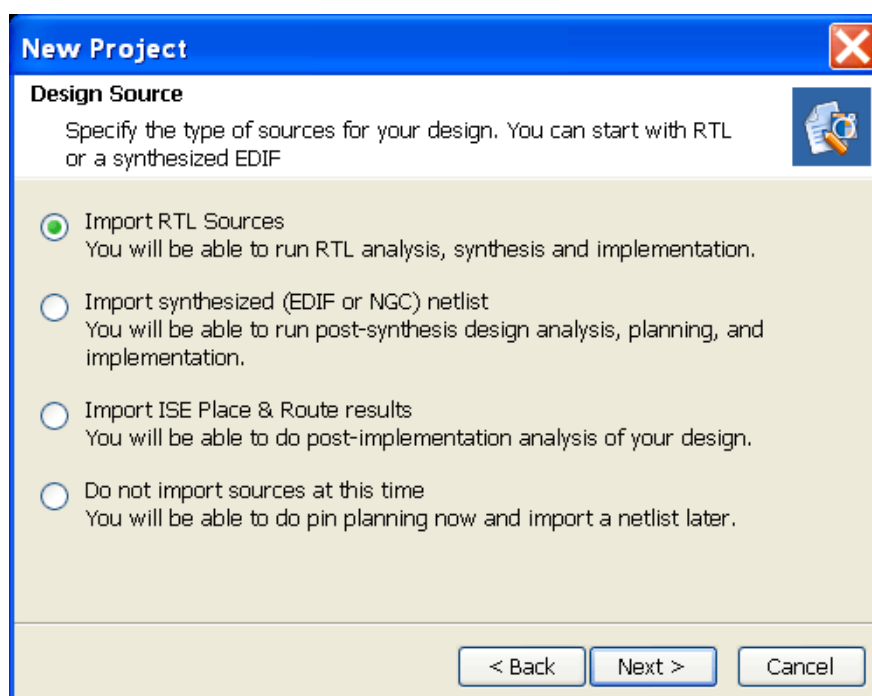


Figure 2-7: New Project Wizard: Design Source Chooser

5. After the above selection is defined, click **Next**.
6. Depending on the desired design input, continue with the instructions in one of the following sections:
 - ◆ “Creating an Empty Project for I/O Pin Planning”
 - ◆ “Creating a Project by Importing RTL Sources”
 - ◆ “Creating a Project with Synthesized EDIF or NGC Format Netlists”
 - ◆ “Creating a Project with ISE Placement and Timing Results”

Note: If you select “Do not import sources at this time” above and you do not import a netlist, the PinAhead pin planning layout will launch when you finish defining the new project.

Creating an Empty Project for I/O Pin Planning

An empty project can be created for I/O pin planning purposes prior to having completed HDL or synthesized EDIF. For more information about I/O pin planning, see [Chapter 5, “I/O Pin Planning.”](#)

1. Select the **Do not import sources at this time** option in the Design Source dialog box.

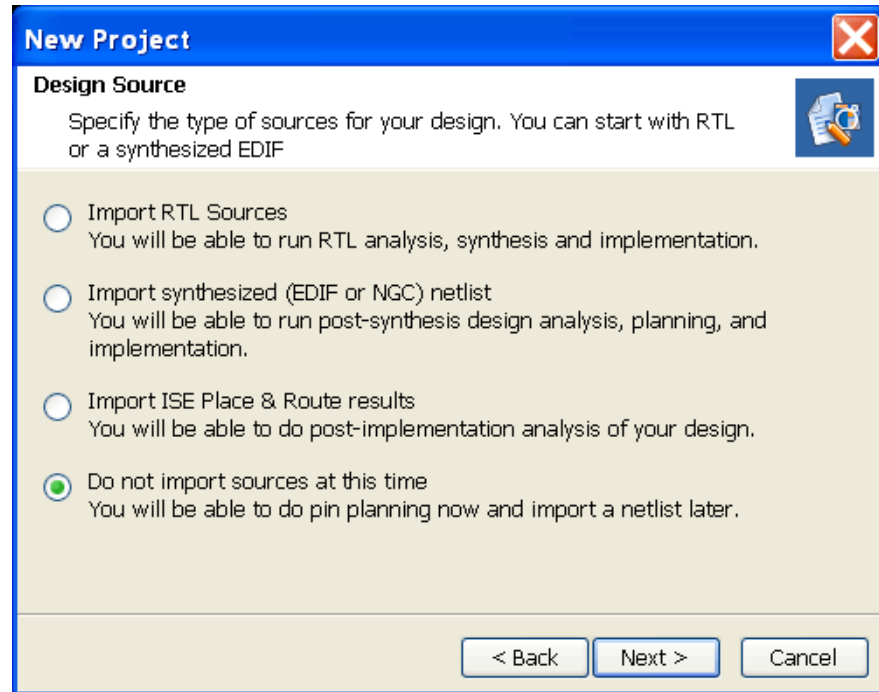


Figure 2-8: Creating an Empty Project

2. Click **Next**.

Selecting a Product Family and Default Part

The next page in the New Project wizard prompts you to select a product family and default part.

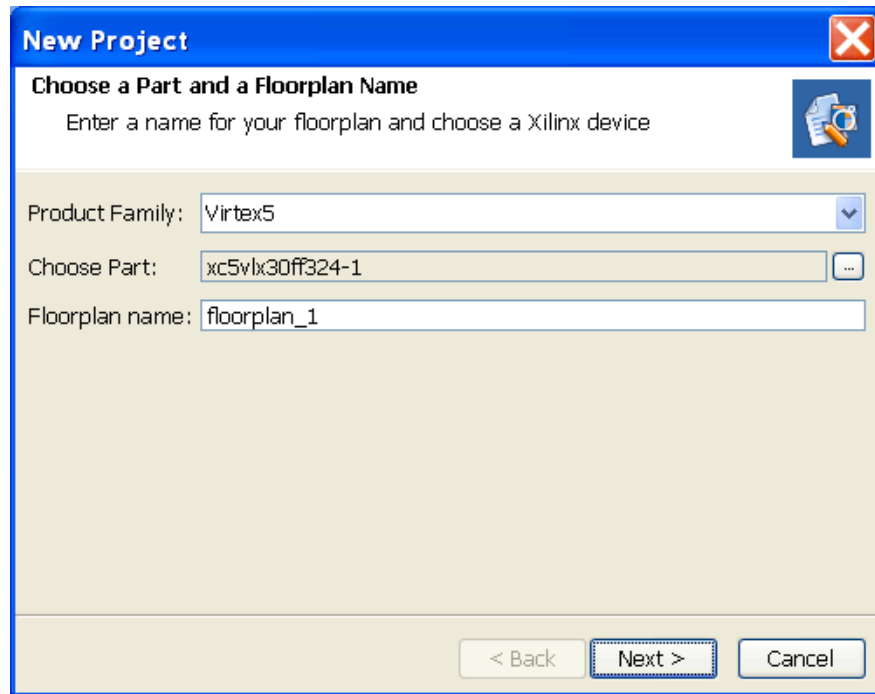


Figure 2-9: New Project Wizard: Product Family and Default Part Page

3. Select the desired target product family architecture and default part.
4. Click **Next**.

Note: Once a Product Family is selected for a Project, it cannot be changed. A new Project will need to be created to target a different architecture. The Default Part, however, can be changed during Synthesis and Implementation Run creation and during Floorplan creation.

Defining the Initial Floorplan Name and Selecting a Target Device

The Floorplan Name page of the New Project wizard is now invoked.

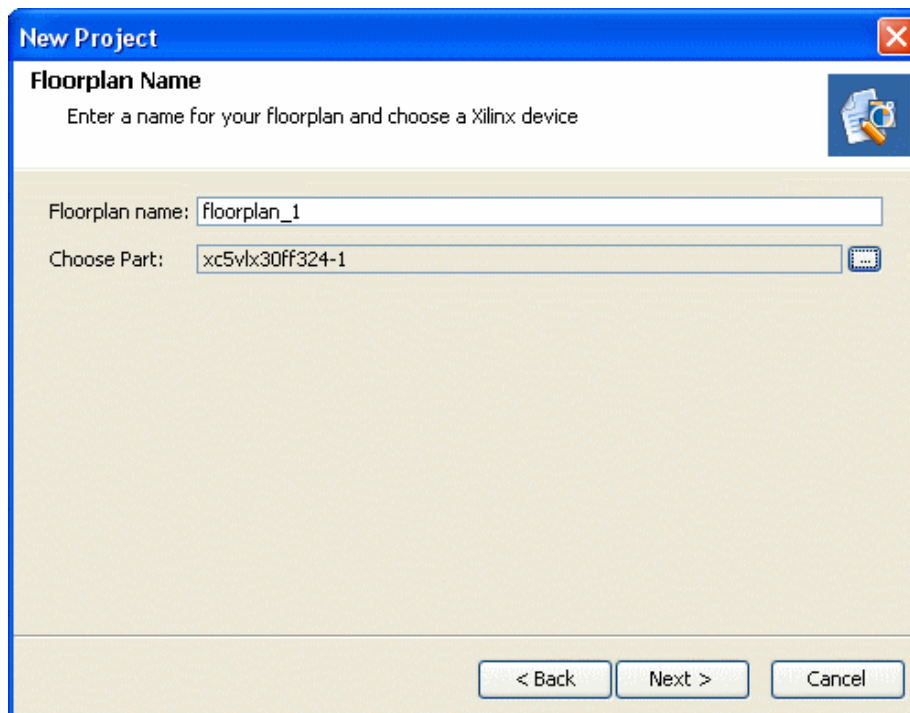


Figure 2-10: New Project Wizard: Floorplan Name Page

5. In the Floorplan Name page, view and edit the editable options.
 - ♦ **Floorplan name**—Enter the desired name for the Floorplan.
 - ♦ **Choose Part**—Use the device browser to select a desired device or accept the default.
6. Click **Next**.

The New Project Summary page is displayed next.
7. To initiate the Project, click **Finish** in the Summary page.

PlanAhead then displays the Project environment with the I/O pin planning related views.

Creating a Project by Importing RTL Sources

RTL source files can be imported to create a Project. This can be used for RTL code development and analysis purposes as well as synthesis and implementation. For more information on RTL development and analysis, see [Chapter 6, “Creating and Analyzing the RTL Design.”](#)

1. Select the **Import RTL sources** option in the Design Source dialog box.

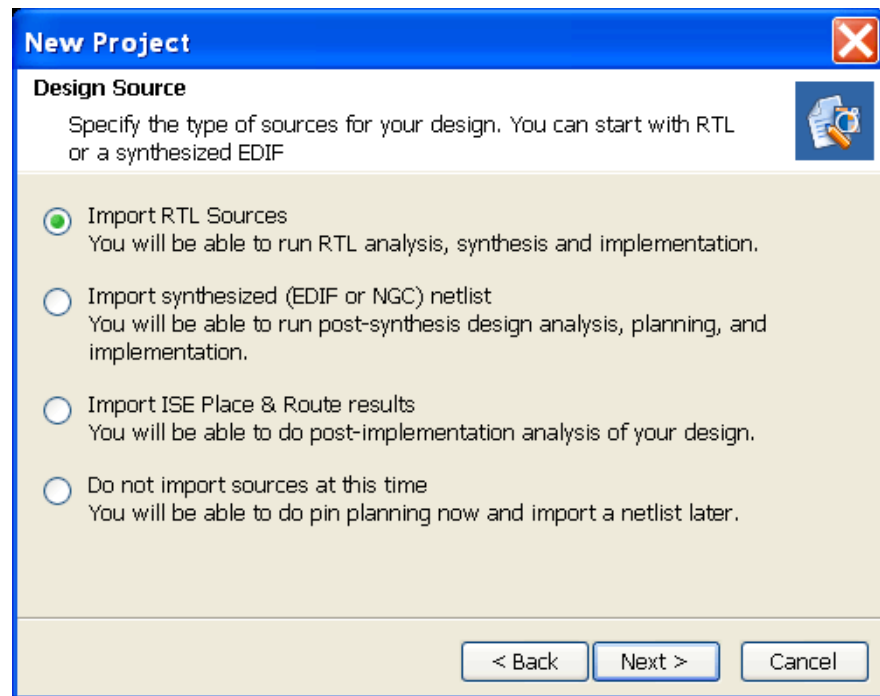


Figure 2-11: Creating a Project with RTL Sources

2. Click **Next**.

Selecting a Product Family and Default Part

The next page in the New Project wizard prompts you to select a product family and default part.

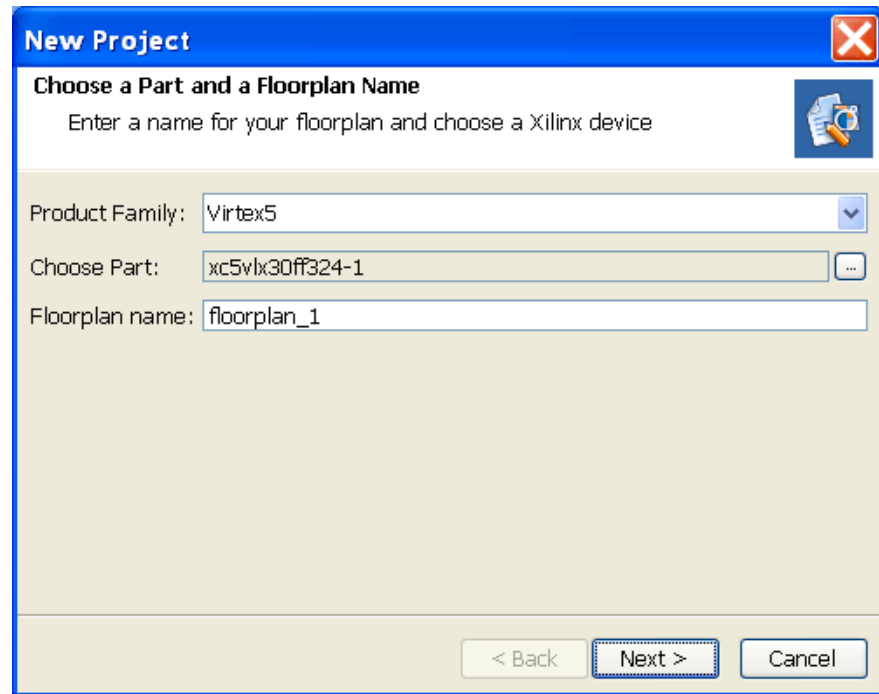


Figure 2-12: New Project Wizard: Product Family and Default Part Page

3. Select the desired target product family architecture and default part.
4. Click **Next**.

Note: Once a Product Family is selected for a Project, it cannot be changed. A new Project will need to be created to target a different architecture. The Default Part, however, can be changed during Synthesis and Implementation Run creation and during Floorplan creation.

Adding Source Files or Directories

The next page in the New Project wizard enables you to select HDL source files or directories that contain HDL source files.

- ◆ **Add Files**—To add individual files to the Project, select the **Add Files** button, and browse to and select the file(s).
- ◆ **Add Directories**—To add the contents of an entire directory and its subdirectories to the Project, select the **Add Directories** button, and browse to and select the directory. All files with source recognized file extensions that are located in the directory tree will be imported.
- ◆ **Import Sources into Project**—To copy the source files into the PlanAhead Project directory structure rather than reference the original locations.

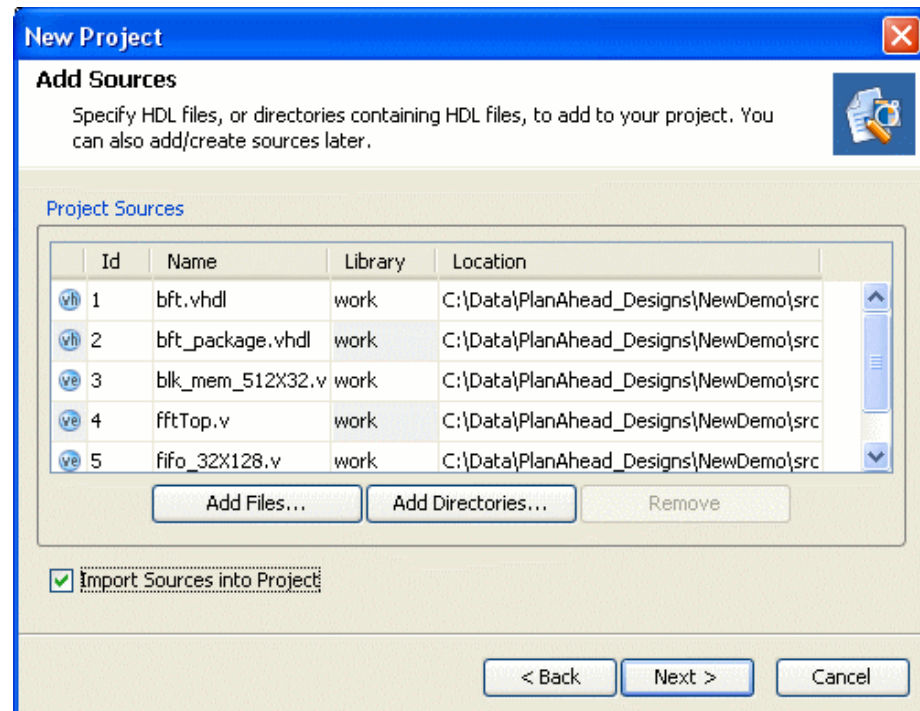


Figure 2-13: New Project Wizard: Add Sources

5. After adding the intended source files or directories, click **Next**.
The HDL Source files are imported into PlanAhead.
The New Project Summary page is displayed next.
6. To initiate the Project, click **Finish** in the Summary page.
PlanAhead then displays the Project environment with the RTL related views available.

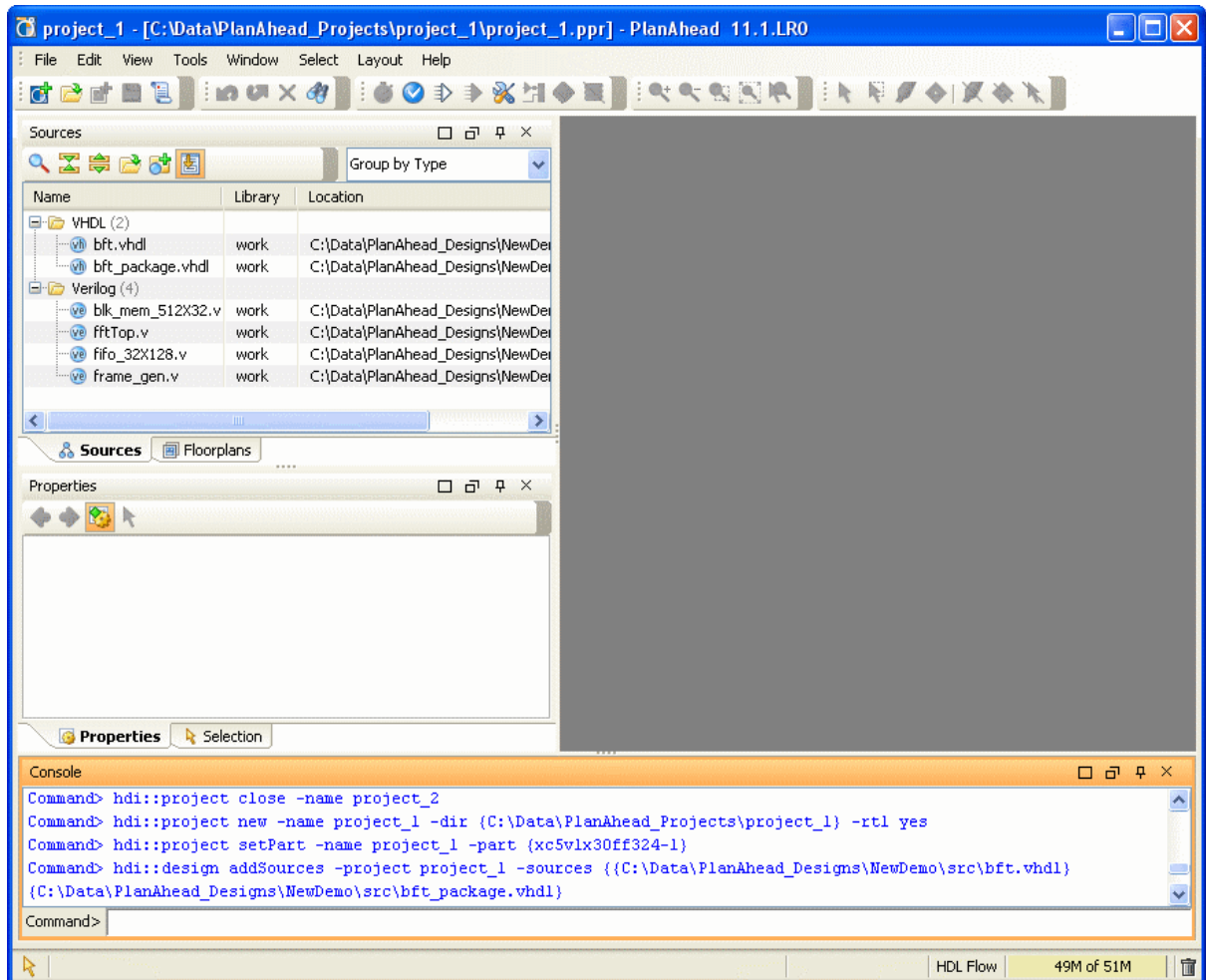


Figure 2-14: PlanAhead RTL Environment

Note: If any warning messages appear in the PlanAhead Console view, it is highly recommended that you examine either these messages or the PlanAhead log file to identify the design errors or specific issues that may cause problems during implementation.

Creating a Project with Synthesized EDIF or NGC Format Netlists

Synthesized netlists along with corresponding constraints can be imported to create a PlanAhead Project. This is used to analyze, floorplan or implement the design using the extensive floorplanning and implementation environment.

1. Select the **Import synthesized (EDIF or NGC) netlist** option in the Design Source dialog box.

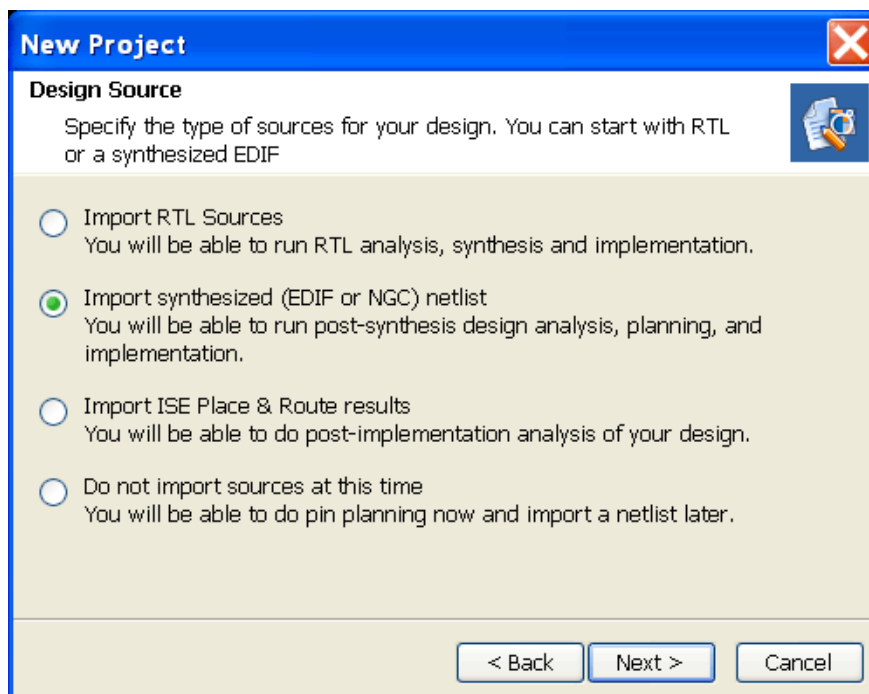


Figure 2-15: Creating a Project using a Synthesized EDIF or NGC Netlist

Selecting a Top-Level Netlist and Module Search Path

If the *Import synthesized (EDIF or NGC) netlist* option was selected, the next page in the New Project wizard enables you to input a top-level netlist file and a search path to find module level netlists.

1. In the Import Netlist page, edit the definable option:
 - ◆ **Netlist file**—Enter a name to identify the top-level netlist in this project. Use the File Browse button to select the top-level netlist file for the design.
 - ◆ **Netlist directories**—Use the Add button to select directories in which to search for lower-level modules and cores during netlist import. By default, the PlanAhead invocation directory and the directory that the top-level netlist was selected from are included in the search path. You can arrange the order in which to search these directories by selecting them and using the up or down arrows buttons. Directories can be removed from the search path by using the Remove button in the dialog box.

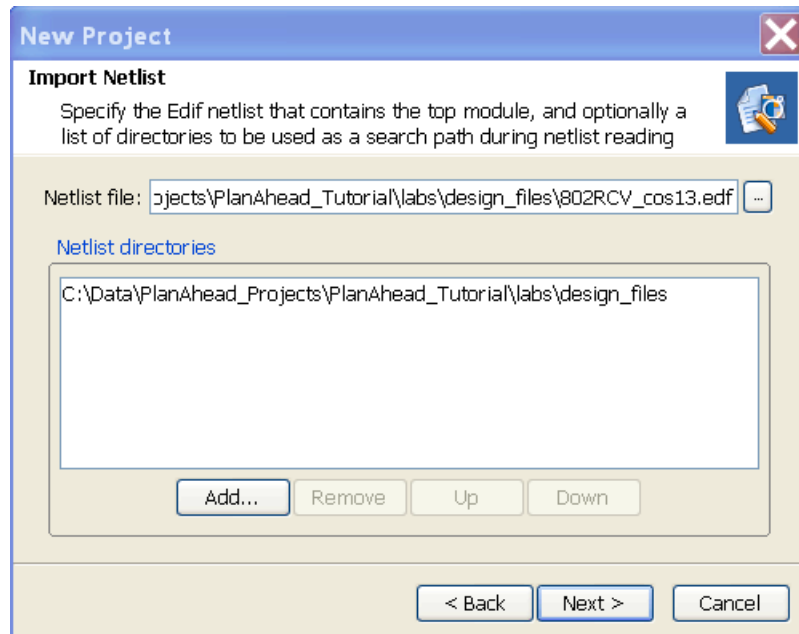


Figure 2-16: New Project Wizard: Import Netlist Page

2. To continue with the wizard, click **Next**.

The netlist will now be imported into PlanAhead which may take a few moments. A status bar is displayed for each netlist imported. Any information regarding warnings and errors will be displayed in the Console view and written to the `planAhead.log` file. A successful file parser message should be displayed.

Note: PlanAhead relies heavily on logic hierarchy for floorplanning. Imported netlists must contain logic hierarchy. Flat netlists are very difficult to floorplan. Ensure that designs are coded in RTL and synthesized hierarchically.

Note: Importing either top-level or module-level NGC or NGO format files requires the use of the `ngc2edif` command to create EDIF files for PlanAhead. The status of the `ngc2edif` commands is displayed in the PlanAhead Terminal window as the commands are running.

Selecting a Product Family, Default Part and Floorplan Name

The next page in the New Project wizard prompts you to select a product family and default part.

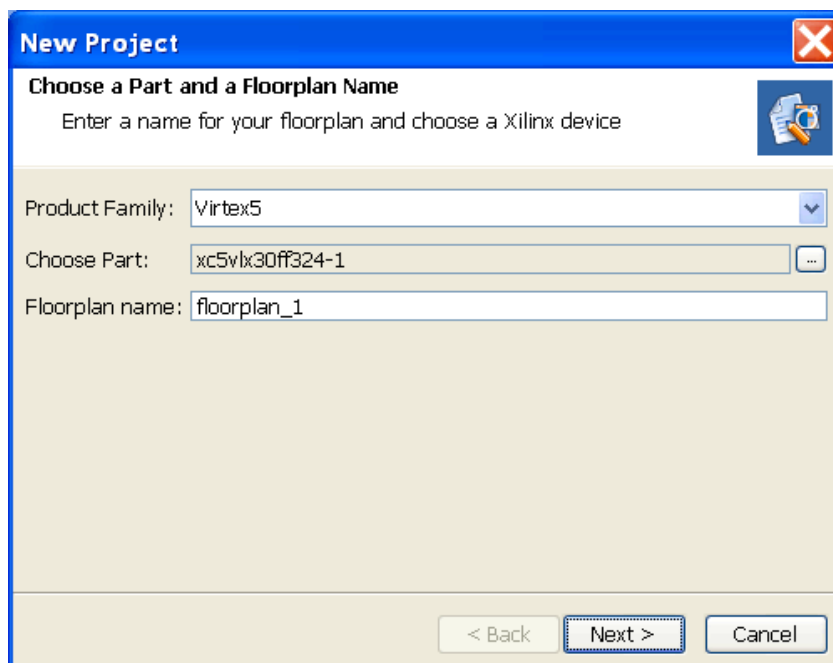


Figure 2-17: New Project Wizard: Product Family and Default Part Page

3. View and edit the definable options:
 - ♦ **Product Family**—Displays the device to be used. Only compatible devices can be selected here.
 - ♦ **Choose Part**—Use the device browser to select a desired device or accept the default to use the entry defined in the top-level EDIF netlist file.
 - ♦ **Floorplan name**—Enter the desired name for the Floorplan.
4. Click **Next**.

Note: Once a Product Family is selected for a Project, it cannot be changed. A new Project will need to be created to target a different architecture. The Default Part, however, can be changed during Synthesis and Implementation Run creation and during Floorplan creation.

Importing Constraints

The Import Constraints page of the New Project wizard is now invoked.

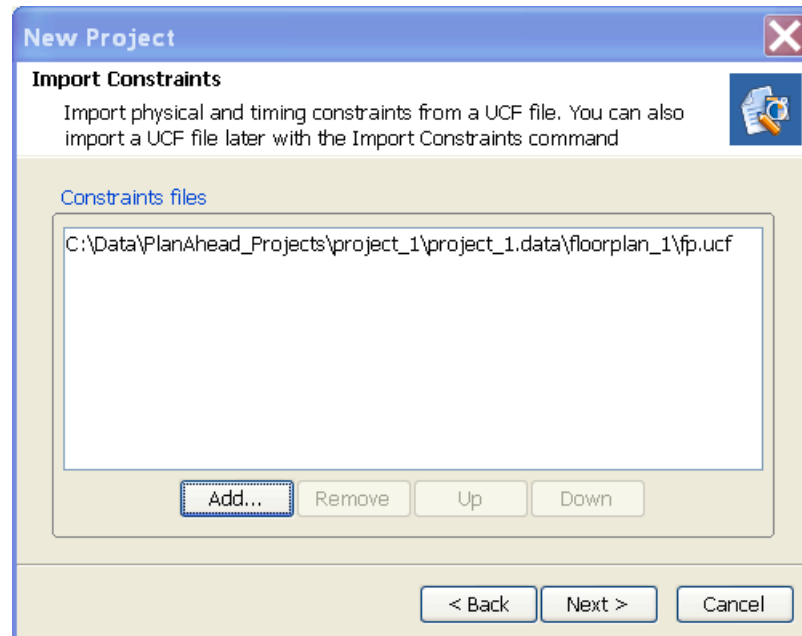


Figure 2-18: New Project Wizard: Import Constraints Page

5. Use the Add button to select *top-level* UCF or NCF constraint files for import. You can arrange the order in which to import these files by selecting them and using the Up or Down buttons. Files can be removed from the list by using the Remove button in the dialog box.

If module-level NCF or UCF constraints are being used, do not include them here. Refer to the [“Importing Constraints” on page 68](#) for more information on importing module-level constraints.

6. To continue the wizard, click **Next**.

The UCF files are imported into PlanAhead. This may take a few moments.

The New Project Summary page is displayed next.

7. To initiate the Floorplan, click **Finish** in the Summary page.

The Floorplan is initialized and the PlanAhead floorplanning environment is invoked.

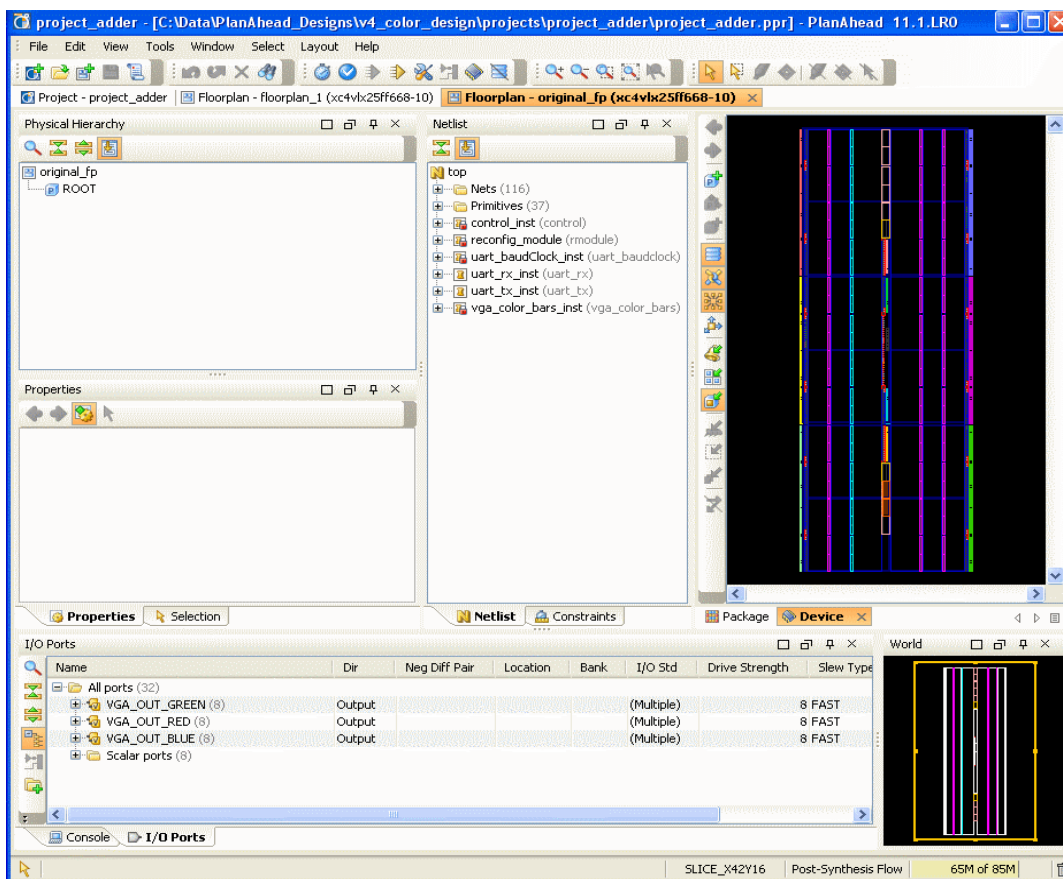


Figure 2-19: PlanAhead Floorplanning Environment

Note: If any warning messages appear in the PlanAhead Console view, examine the messages or the PlanAhead log file to identify the design errors or specific issues that may cause problems during implementation.

Creating a Project with ISE Placement and Timing Results

Netlists with ISE® implementation results along with corresponding constraints can be imported to create a PlanAhead Project. This is used to analyze the place and route results using the implementation environment.

Select the **Import ISE Place & Route results** option in the Design Source dialog box.

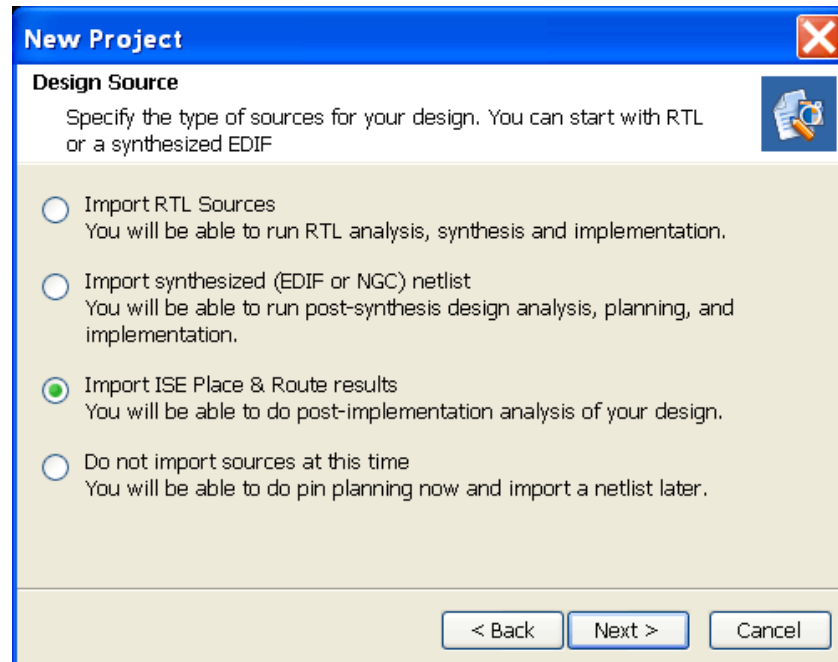


Figure 2-20: Creating a Project with ISE Place and Route Results

Selecting a Top-Level Netlist and Module Search Path

If the *Import ISE Place & Route results* option was selected, the next page in the New Project wizard enables you to input a top-level netlist file and a search path to find module level netlists.

1. In the Import Netlist page, edit the definable option:
 - ◆ **Netlist file**—Enter a name to identify the top-level netlist in this project. Use the File Browse button to select the top-level netlist file for the design.
 - ◆ **Netlist directories**—Use the Add button to select directories in which to search for lower-level modules and cores during netlist import. By default, the PlanAhead invocation directory and the directory that the top-level netlist was selected from are included in the search path. You can arrange the order in which to search these directories by selecting them and using the Up or Down buttons. Directories can be removed from the search path by using the Remove button in the dialog box.

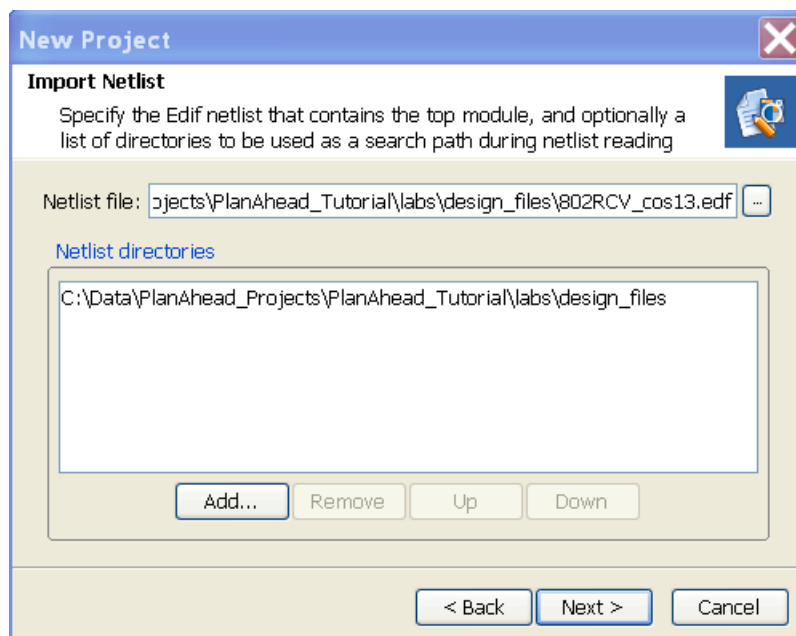


Figure 2-21: New Project Wizard: Import Netlist Page

2. To continue with the wizard, click **Next**.

The netlist will now be imported into PlanAhead which may take a few moments. A status bar is displayed for each netlist imported. Any information regarding warnings and errors will be displayed in the Console view and written to the `planAhead.log` file. A successful file parser message should be displayed.

Selecting a Product Family and Default Part

The next page in the New Project wizard prompts you to select a product family and default part.

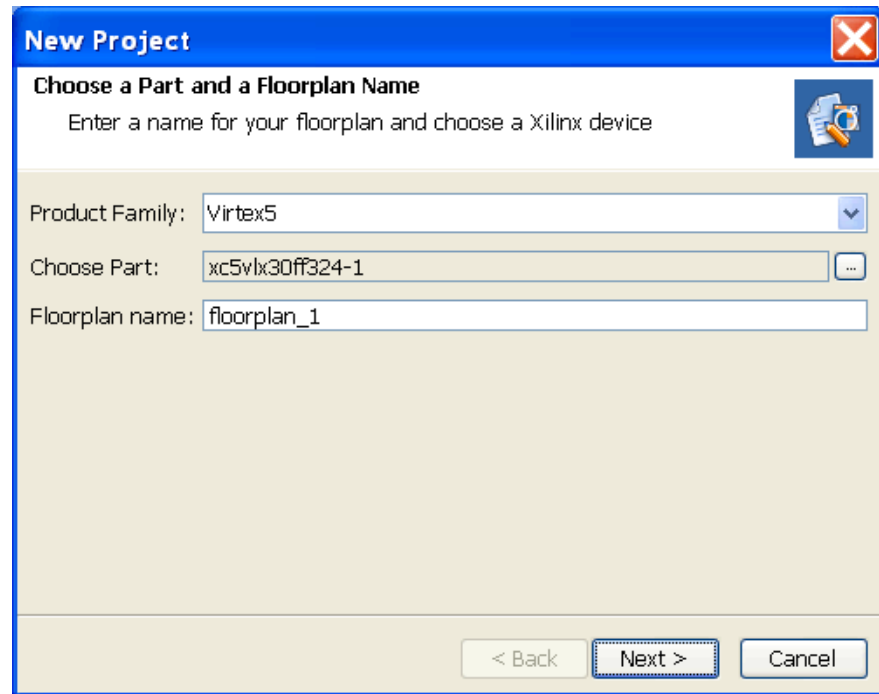


Figure 2-22: New Project Wizard: Product Family and Default Part Page

3. Select the desired target product family architecture and default part.
4. Click **Next**.

Note: Once a Product Family is selected for a Project, it cannot be changed. A new Project will need to be created to target a different architecture. The Default Part, however, can be changed during Synthesis and Implementation Run creation and during Floorplan creation.

Defining the Initial Floorplan Name and Selecting a Target Device

The Floorplan Name page of the New Project wizard is now invoked.

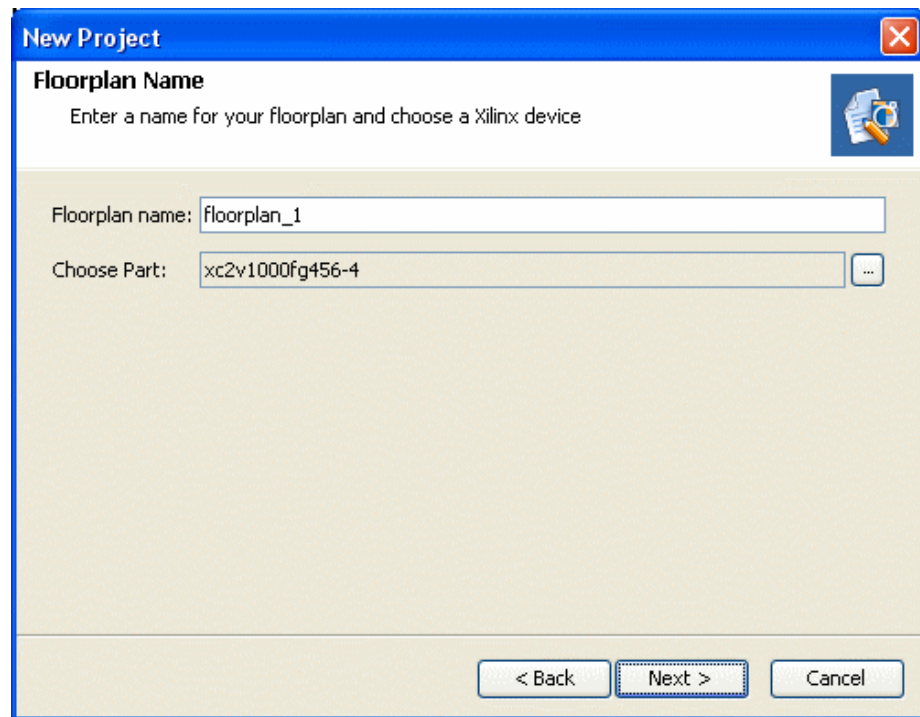


Figure 2-23: New Project Wizard: Floorplan Name Page

5. In the Floorplan Name page, view and edit the definable options:
 - ♦ **Floorplan name**—Enter the desired name for the Floorplan.
 - ♦ **Choose Part**—Use the device browser to select a desired device or accept the default to use the entry defined in the top-level EDIF netlist file.
6. Enter a floorplan name, and select the target device.
7. Click **Next**.

Importing Constraints

The Import Constraints page of the New Project wizard is now invoked.

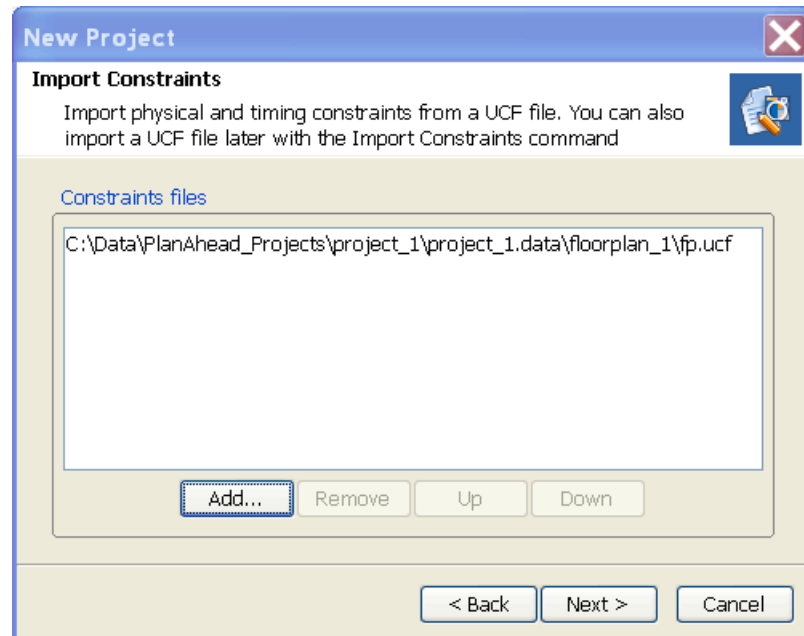


Figure 2-24: New Project Wizard: Import Constraints Page

8. Use the Add button to select *top-level* UCF or NCF constraint files for import. You can arrange the order in which to import these files by selecting them and using the Up or Down buttons. Files can be removed from the list by using the Remove button in the dialog box.

If module-level NCF or UCF constraints are being used, do not include them here. Refer to the [“Importing Constraints” on page 68](#) for more information on importing module-level constraints.

9. To continue the wizard, click **Next**.

The UCF files are imported into PlanAhead. This may take a few moments.

Importing Placement and Timing Results

The Import ISE Implementation Results page is invoked. It enables you to import the place and route results generated in ISE which are used to create a floorplan for viewing and analysis in PlanAhead.

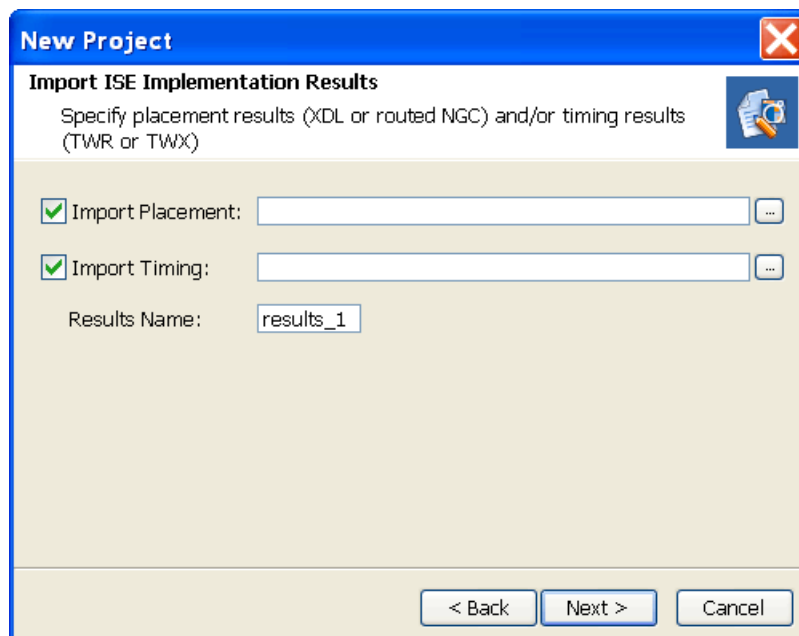


Figure 2-25: New Project Wizard: Import ISE Implementation Results

10. In the Import ISE Implementation Results page, edit the definable options:
 - ♦ **Import Placement**— Select this option, and locate and select a placement results file from the ISE implementation , such as an XDL, GZ, or NCD format file.
 - ♦ **Import Timing**—Select this option, and locate and select a timing results file from the ISE implementation, such as a TWX format file.
 - ♦ **Results Name**—Enter the name of the results floorplan.

The New Project Summary page is displayed next.

11. To initiate the Floorplan, click **Finish** in the Summary page.

The Floorplan is initialized and the PlanAhead floorplanning environment is invoked.

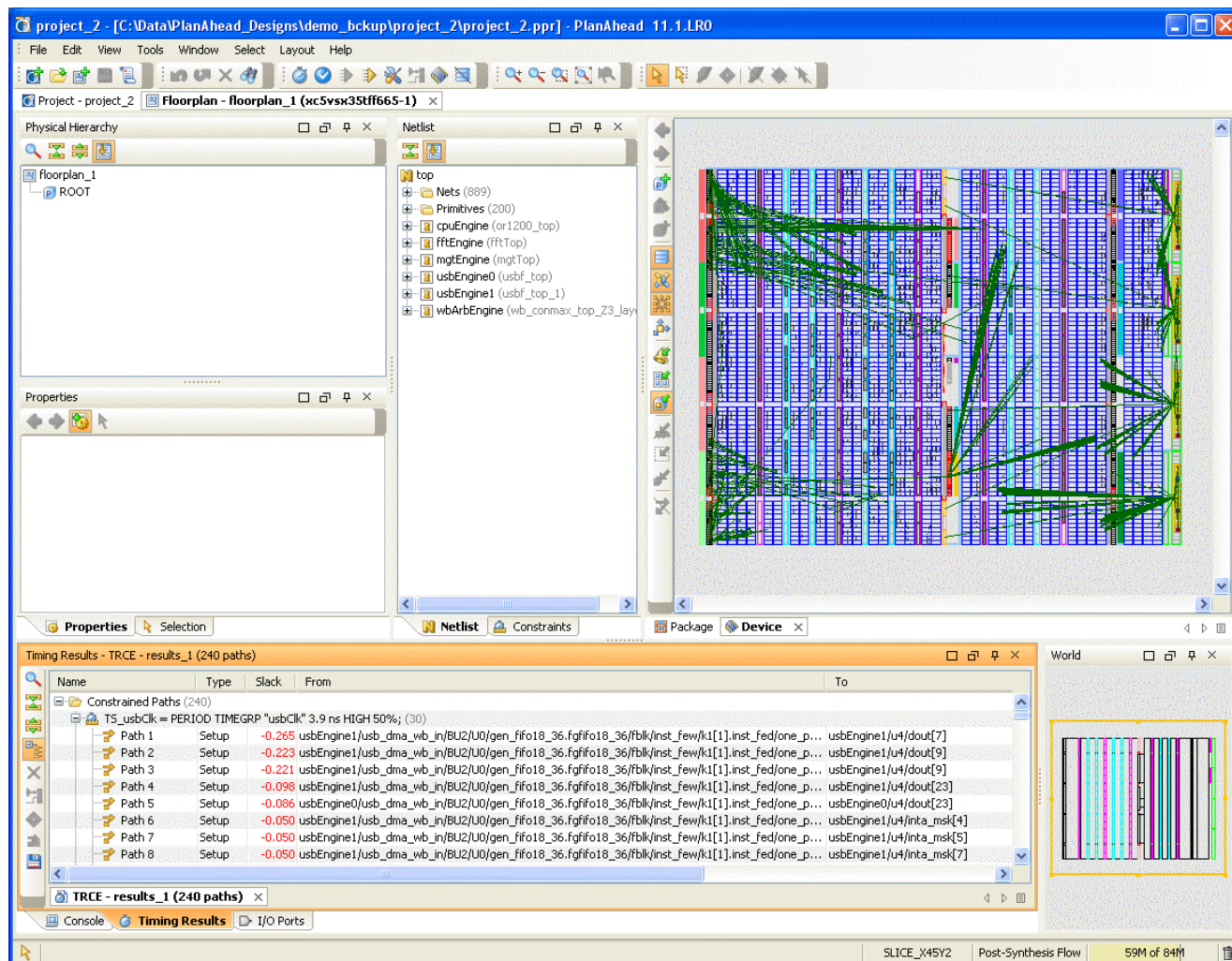


Figure 2-26: PlanAhead Floorplanning Environment

Note: If any warning messages appear in the PlanAhead Console view, examine the messages or the PlanAhead log file to identify the design errors or specific issues that may cause problems during implementation.

Managing Projects

The project maintenance commands described below assume a synthesized netlist was imported to create the Project. Some of the commands are not available or applicable for PinAhead Projects or Projects created with HDL sources.

Opening an Existing Project

Existing Projects are opened in the PlanAhead software by using the Open Project command, or in Windows, by double-clicking any PlanAhead project (PPR) file, which invoked PlanAhead and open the project. As Projects are opened, the project “state” from the time the Project was last closed is restored. The state of all previous Implementation runs is restored. Previously opened Floorplans are re-opened and available for modification.

The applicable view layout is displayed for the type of Project being opened or created--PinAhead, HDL Sources or PlanAhead with a synthesized netlist.

To open a Project in PlanAhead, use one of the following methods:

- Select **File > Open Project**
- Click the Open Project toolbar button.



Figure 2-27: Open Project Toolbar Button

- Click the **Open a Previously Created Project** link in the Getting Started jump page. The Open Project browser is displayed. Select a PPR project file to open.

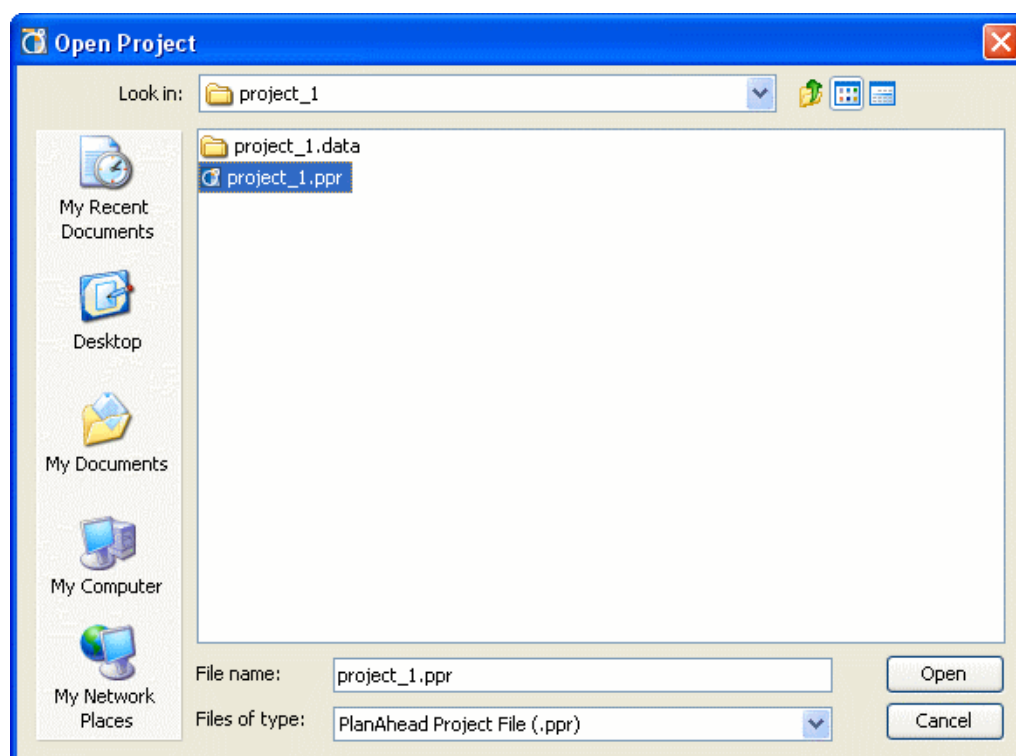


Figure 2-28: Open Project Dialog Box

Opening Multiple Projects

Multiple Projects can be opened simultaneously in a single PlanAhead session by using any of the methods described above in the Opening Existing Projects section of this chapter. A separate PlanAhead main window will appear for each opened project. Some window focus issues may present themselves with multiple projects opened simultaneously.

System memory requirements may hinder performance when opening multiple projects. You can monitor the Java memory consumption status bar at the bottom right corner of the PlanAhead environment window. PlanAhead has a preset Java memory limit 512MB on Windows, and 1GB on 64-bit Linux. If you get close to that number, you may wish to close Projects or Floorplans in order to reduce the memory being used.

Closing a Project

Projects can be closed from within PlanAhead with the **File > Close Project** command. Closing a Project will prompt you to save any unsaved Floorplans. You may elect to exit without saving the data, or you may choose to save the data. Netlist updates are automatically saved in the Project as they are executed.

Updating the Netlist for a Synthesized Netlist Based Project

Updating the Top-Level Netlist

To update an existing Project with a newly synthesized netlist for the entire design, do as follows.

1. Select **File > Update Netlist**.

The Update Netlist wizard will appear.

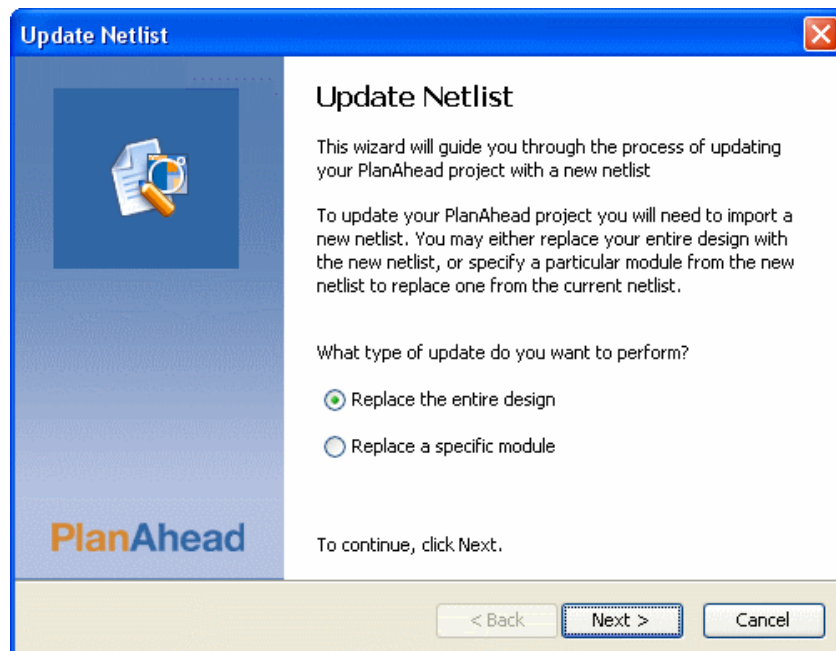


Figure 2-29: Update Netlist Wizard

2. Select the **Replace the entire design** option from the Update Netlist dialog box to update the entire design netlist.
3. Click **Next** to continue.
The Import Netlist dialog box appears.
4. Set the following two editable options in the Import Netlist dialog box.
 - ♦ **Netlist file**—Enter a name to identify the top-level netlist in this project. Use the File Browser to select the top-level netlist file for the design.
 - ♦ **Netlist directories**—Use the Add button to select directories to be searched for lower-level modules and cores during netlist import. By default, the PlanAhead invocation directory and the directory that the top-level netlist was selected from are included in the search path. You can arrange the order in which to search these

directories by selecting them and using the Up or Down buttons. Directories can be removed from the search path by using the Remove button.

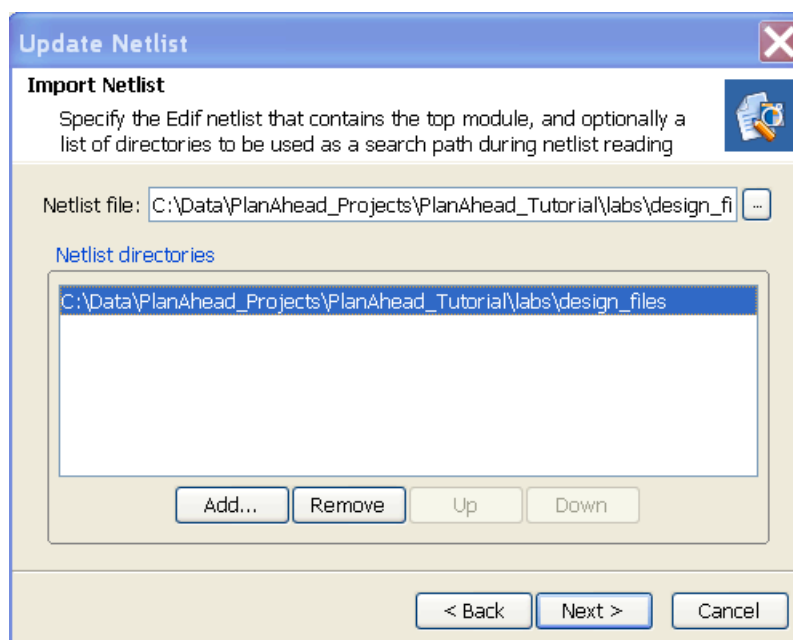


Figure 2-30: Update Netlist Wizard: Import Netlist Dialog Box

5. To continue, click **Next**.
6. In the Update Netlist Summary dialog box, click **Finish** to continue.

The netlist will now be imported into PlanAhead which may take a few moments. A status bar is displayed for each netlist imported. Any information regarding warnings and errors will be displayed in the Console view and written to the `planAhead.log` file. A successful file parser message should be displayed.

Updating a Module-Level Netlist

The PlanAhead Update Netlist command enables you to import either a new module-level or top-level netlist as the basis for module replacement. You can select any module contained in the newly imported netlist to replace any module in the existing project. The only caveat is that the module port interface remains the same. This often requires certain synthesis parameters to ensure that logic is not optimized across hierarchical logic module boundaries. Refer to the appropriate synthesis documentation for more information on preserving hierarchy.

PlanAhead can import module-level incremental netlists from synthesis. This capability enables designers to implement an incremental synthesis strategy as well.

Note: When a design is re-synthesized, lower level net and instance names often change. This causes a mismatch if the Floorplan contains instance level constraints such as LOCs. To avoid potential problems during update, PlanAhead will unload the module constraints prior to updating the netlist. After the update, the constraints can be re-imported and those that match will get assigned. This usually is only an issue if placement constraints are defined in the design. It is a good idea to remove module constraints before updating the netlist.

Netlists can then be incrementally updated by using the Update Netlist command to import individual modules of the design. To do so:

1. Select **File > Update Netlist**.

The Update Netlist wizard will appear.

2. Set the **Replace a specific module** option in the Update Netlist dialog box to selective update of modules in the design.

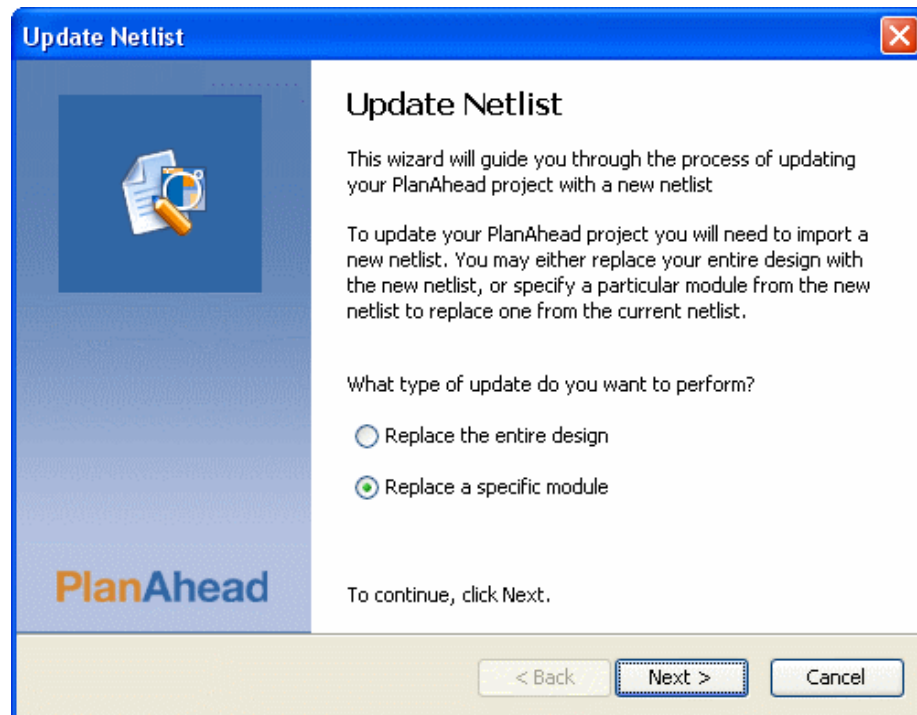


Figure 2-31: Update Netlist Wizard

3. Click **Next** to continue.
The Import Netlist dialog box appears.

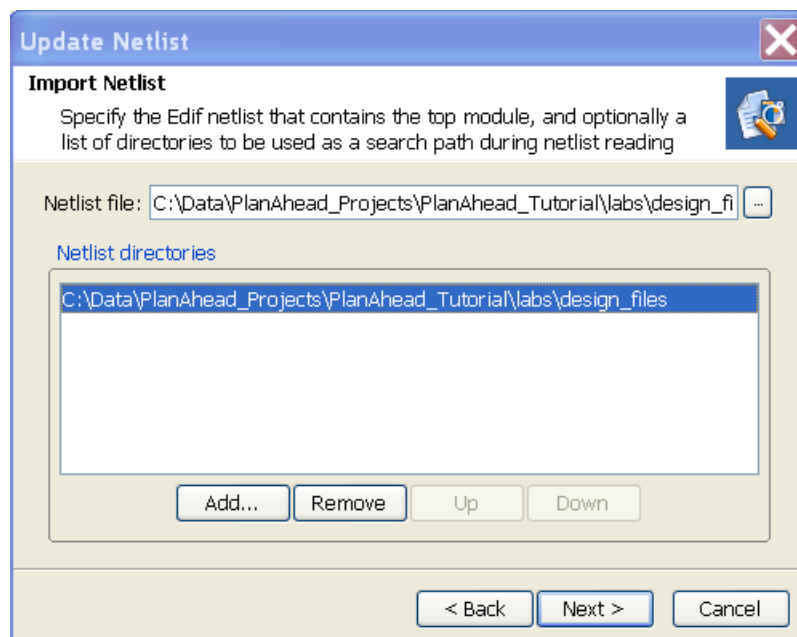


Figure 2-32: Update Netlist Wizard: Import Netlist Dialog Box

The Import Netlist dialog box contains the following editable options:

- ♦ **Netlist file**—Enter a name to identify the netlist containing the module to update the project. Use the file browser to select the module netlist file for the design.
 - ♦ **Netlist directories**—Use the Add button to select directories to be searched for lower-level modules and cores during netlist import. By default, the PlanAhead invocation directory and the directory that the top-level netlist was selected from are included in the search path. You can arrange the order in which to search these directories by selecting them and using the Up or Down buttons. Directories can be removed from the search path by using the Remove button.
4. Click **Next** to continue.

The Specify Replacement Module dialog box will appear.

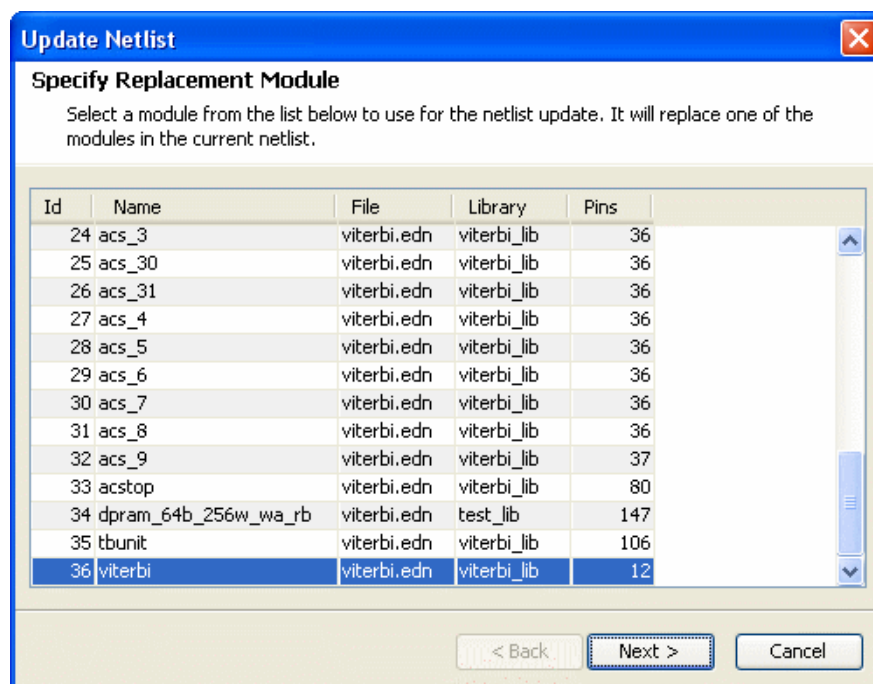


Figure 2-33: Update Netlist Wizard: Specify Replacement Module Page

5. Select the module to be used as the replacement during the update from the list. By default, PlanAhead will automatically select the top-level module in the replacement netlist. The list can be sorted by clicking on the column headers.
6. Click **Next** to continue.
- The Specify Module to be Replaced dialog box will appear.
7. Select the module from the original netlist to be updated. PlanAhead will attempt to select the module to be replaced based on the name of the replacement module. You may need to scroll to find the module you want to select.
8. Click **Next** to continue.
- The Summary dialog box will appear.
9. Review the Summary for the module replacement.
10. Click **Finish** to update the design with the new module-level netlist.

Working with Floorplans

What is a Floorplan?

PlanAhead provides the Floorplanning environment to analyze and experiment with various design parameters including alternate devices, timing and/or physical constraints and implementation options. Think of a floorplan as a snapshot of design constraints targeted at a specified netlist and device. They are created in PlanAhead to modify or experiment with devices or constraints.

You can select any number of top-level UCF files to be used during creation of the floorplan. Module level UCF files can also be imported using the Import Constraints command. Once imported, PlanAhead manages all of the constraints within the floorplan. The original imported UCF files are no longer referenced by PlanAhead or by the implementation tools. All of the imported and newly assigned constraints are merged together into a single UCF file passed to the implementation tools.

Multiple floorplans can be created for any given netlist-based project or for any Synthesis run in an RTL based project.

Creating a Floorplan

Synthesized Netlist and Empty Projects

If working with synthesized netlist based or empty pin planning Projects, an initial floorplan is created during the project creation process. You are prompted to define a floorplan name and target device as well as any number of top-level UCF files in the New Project wizard. Module level UCF files can also be imported using the Import Constraints command.

To create a new Floorplan in an existing Project, use of the following methods:

1. Launch the New Floorplan Wizard using one of the following methods:
 - ♦ Select **File > New Floorplan**.
 - ♦ Click the **New Floorplan** toolbar button.



Figure 2-34: New Floorplan Toolbar Button

The New Floorplan dialog box will appear.

2. Click **Next** to continue.

Defining the Floorplan Name and Selecting a Target Device

The Floorplan Name dialog box of the New Floorplan wizard is now invoked.

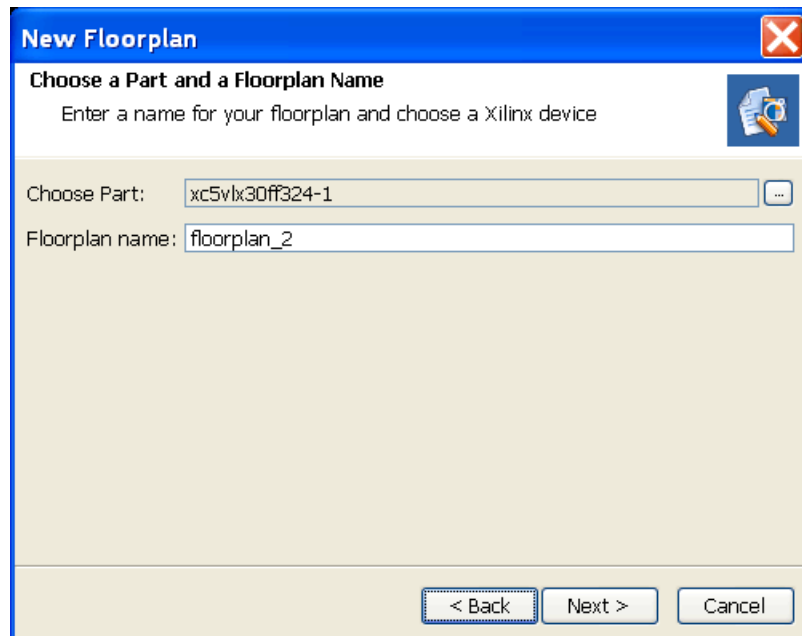


Figure 2-35: New Floorplan Wizard: Floorplan Name and Device

3. Set the following editable options in the Floorplan Name dialog box.
 - ◆ **Floorplan name**—Enter the desired name for the floorplan.
 - ◆ **Device to use:**
 - **From Netlist**—Select to use the entry defined in the top-level EDIF netlist file.
 - **Select Part**—Use the files browser to invoke the Select Part chooser which enables you to select a desired device.
4. Click **Next** to continue.

Importing Constraints

The Import Constraints dialog box of the New Project wizard is now invoked.

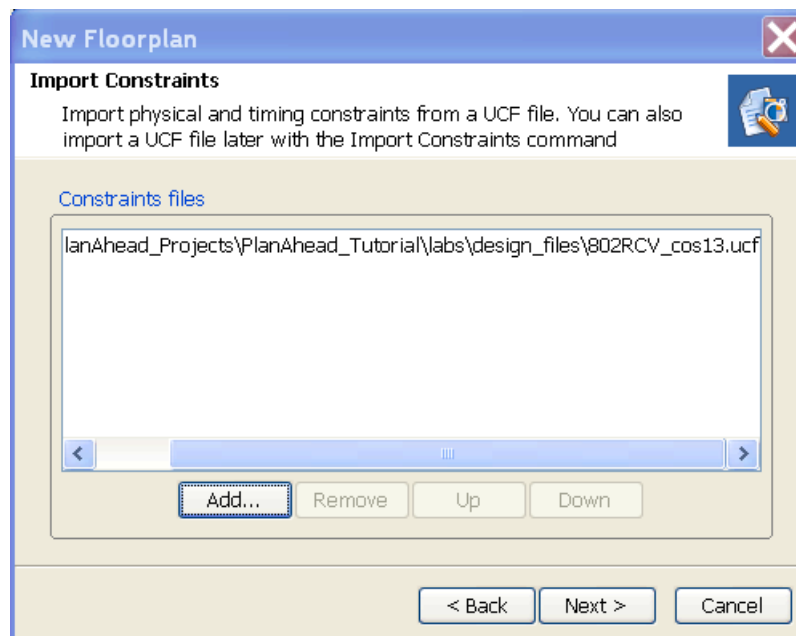


Figure 2-36: New Floorplan Wizard: Import Constraints

1. Click the **Add** button to locate and select *top-level* UCF or NCF constraint files for import. You can arrange the order in which to import these files by selecting them and using the Up or Down buttons. Files can be removed from the list by using the Remove button.

If module level constraints are being used, do not include them here. For more information on importing module-level constraints, see [“Importing Module-Level Constraints.”](#)

2. Click **Next** to continue.

The UCF files are imported into PlanAhead. This may take a few moments.

The New Floorplan Summary dialog box is now displayed.

3. To initiate the floorplan, click **Finish**.

The new Floorplan is initiated and active in the PlanAhead environment.

Additional Floorplans can be created using the **New Floorplan** or **Copy Floorplan** commands.

RTL Projects

RTL Projects do not require a Floorplan to be created in order to create and launch synthesis and implementation runs. As implementation runs are launched, a target device and single top-level UCF can be defined. You can experiment with any number of devices, constraints and implementation options. The one limitation is that only a single UCF file can be assigned to each run, and constraints are not modified by PlanAhead. If you intend to modify constraints or analyze implementation results, a Floorplan is created.

When you use the **Select > Import Run** command to import and analyze implementation results, you are prompted to define a Floorplan name. The Floorplan is then created for you, and results are displayed. The UCF file used to launch the run is loaded into the Floorplan.

A Floorplan can also be created prior to implementation to analyze the design or to modify constraints. Click **Select > Import Run** to import the desired synthesis run for floorplanning. Once loaded, use the New Floorplan command to create a Floorplan. Refer to the [“Creating Multiple Floorplans”](#) for more information.

Additional Floorplans can be created using the **New Floorplan** or **Copy Floorplan** commands.

Importing Constraints

UCF and NCF format constraints files are created by synthesis software tools, designers, Xilinx ISE software tools or by previous sessions of PlanAhead. These files can be imported into a Floorplan. Constraints consist of Timing constraints, I/O locations and standards, and physical location constraints for various instances. For more information regarding UCF format files, refer to the *Xilinx Constraints Guide*.

The Floorplan must be initialized before importing physical constraints. To import physical constraints, do as follows:

1. Select **File > Import Constraints**.

This displays the Import Constraints dialog box.

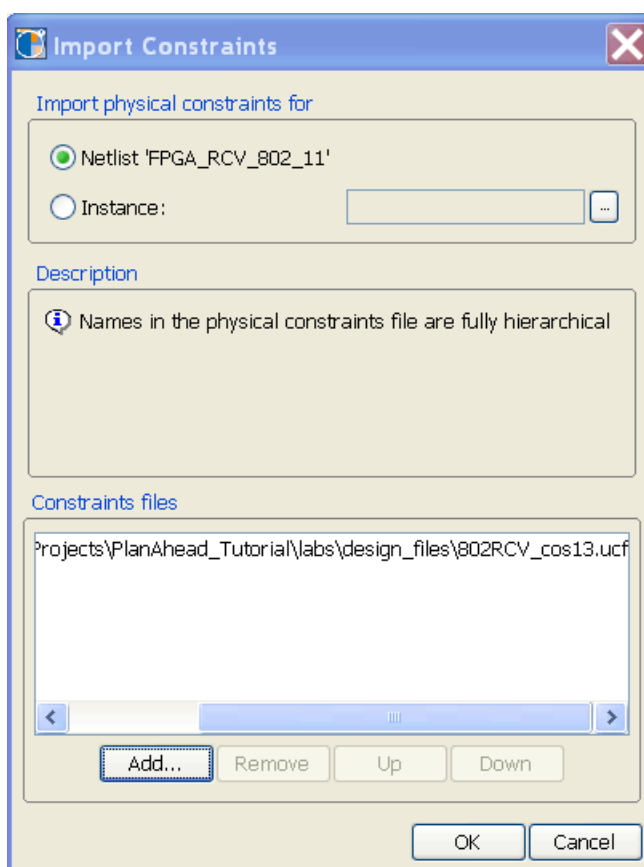


Figure 2-37: Import Constraints Dialog Box

2. Set the following editable options in the Import Constraints dialog box.
 - ♦ **Import physical constraints for**—Selects the level to assign constraints to.
 - **Netlist <netlist name>** imports top-level constraints.
 - **Instance: *Selected instance*** imports constraints for a selected instance. The dialog box is seeded with a pre-selected instance prior to running the command. This button is used when importing module level constraints such as core-level NCF files.
 - ♦ **Constraints files**—Displays the constraints files to import.
3. Click the **Add** button to locate and select the constraint files to import.
4. Click **OK** to import the constraints.

Any information regarding warnings and errors during the import are displayed in the Console view and written to the `planAhead.log` file.

Multiple constraints files can be imported to allow for separation of different types of constraints, such as I/O Port assignments and timing constraints.

Note: Timing constraints will be duplicated if imported more than once. To remove duplicated timing constraints, delete the duplicate in the Constraints view.

Importing Module-Level Constraints

PlanAhead allows selective assignment of module-level constraints.

To assign constraints to a module instance:

1. Select the instance in the Netlist view.
2. Select **File > Import Constraints**.

The Import Constraints dialog box will be displayed.

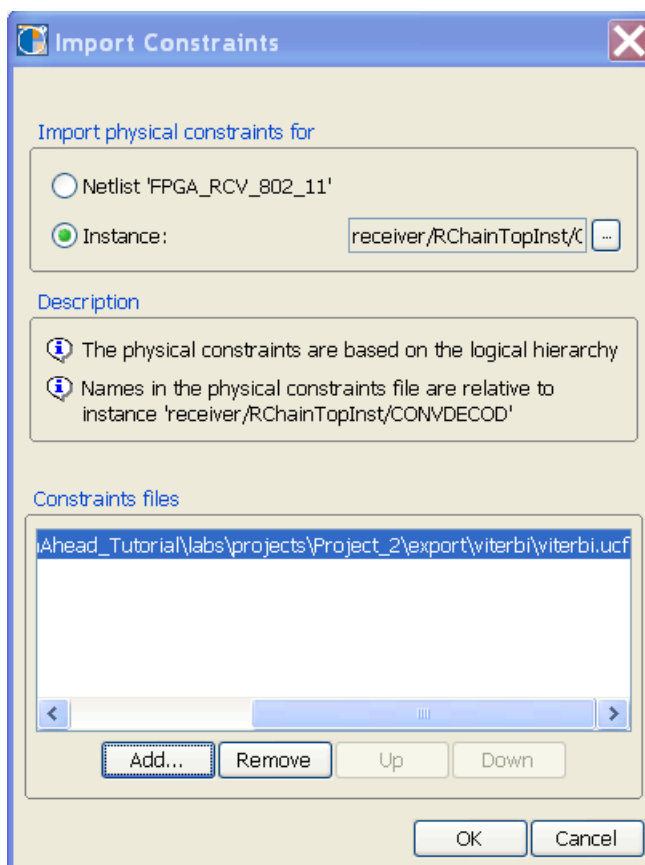


Figure 2-38: Import Constraints Dialog Box

3. Select **Instance** and make sure the “Instance” field is set to the netlist instance for which you are importing module-level constraints.
4. In the “Constraints files” area, click the **Add** button to locate and select the constraint files to import.
5. Click **OK** to import the constraints.

Any information regarding warnings and errors during the import are displayed in the Console view and written to the `planAhead.log` file.

Updating Design Constraints for Floorplans

PlanAhead stores each Floorplan with a unique set of UCF constraint file(s). Currently, all UCF modifications performed in PlanAhead are being performed on the active Floorplan only.

Note: Importing a new UCF file will not automatically update previous UCF settings. To avoid duplicate entries, the constraints being updated should first be removed from the Project before importing the new files. The Constraints view will display any duplicates allowing for removal later.

Removing Timing Constraints

To remove the timing constraints, first configure the Constraints view to list all timing constraints defined. Select the desired constraints to remove and use the Delete command to remove them.

1. Select the **Group by type** toolbar icon in the Constraints view to expand the list of Constraints.
2. Select some or all constraints using **Ctrl+click** or **Ctrl+A**.
3. Select the **Delete** popup menu command.
4. Click **OK** to confirm removal.

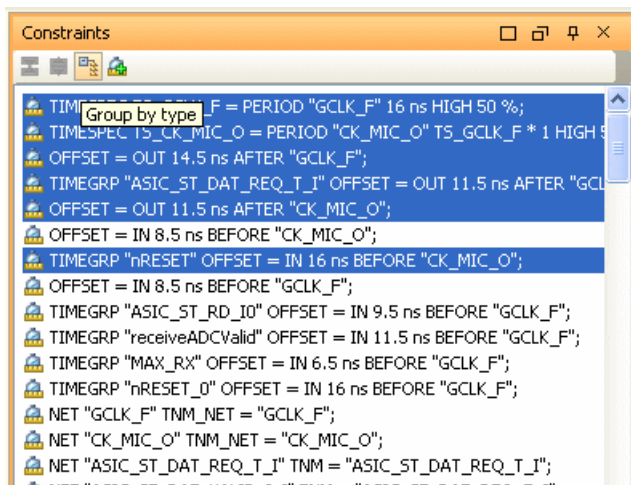


Figure 2-39: Deleting Timing Constraints

Removing Placement Constraints

Placement Constraints assignment can be updated by simply importing new constraints. Previous LOC constraint assignments are replaced with the new assignment. LOC constraints are not removed if they are absent from the constraint file being imported.

Placement constraints can be selectively removed prior to import. For more information, see [“Selectively Clearing Placement Constraints.”](#)

Importing New UCF Constraint Files

New top-level or module-level UCF or NCF format constraint files can be imported.

1. Select the **File > Import Constraints** command.

The *Import physical constraints for* options include:

- ♦ **Netlist ‘netlist name’**—Imports constraints relative to the top-level of the design.
- ♦ **Instance**—Provides a browser to select a logic module instance to import the constraints relative to. A pre-selected logic instance is automatically seeded into the dialog box.

The Constraints files lists all files that have been selected to be imported into the Floorplan. The Add buttons enables you to select UCF files for import. Constraint files are imported in the order they appear in the list. Use the Up or Down buttons to reorder the constraints files for import. Files can be removed from the list by using the Remove button.

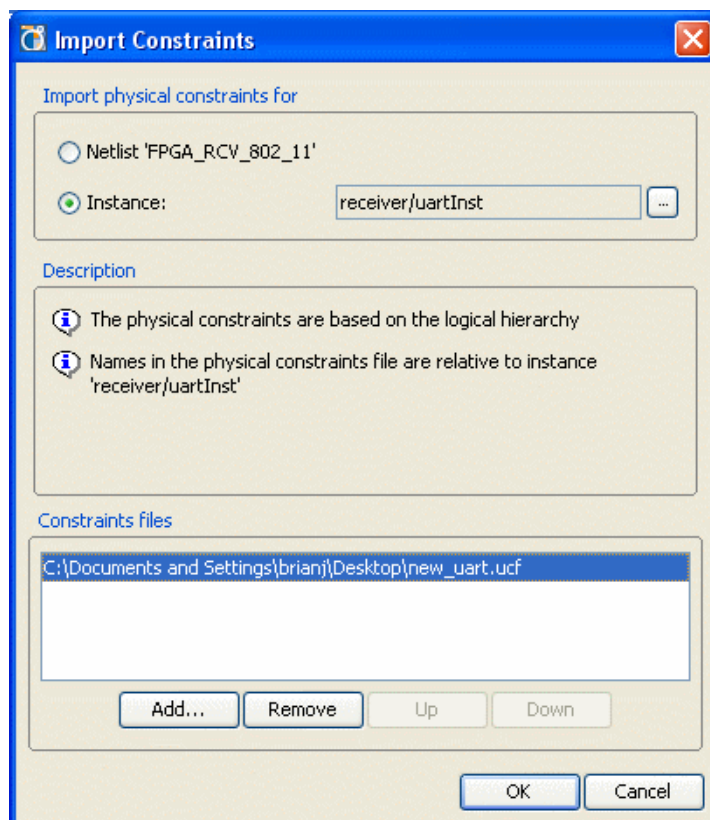


Figure 2-40: Importing New UCF Constraint Files

2. Select either the **Netlist** or **Instance** option to import top-level or module level constraints
3. Use the **Add** browser to select the files to import.
4. When done, click **OK**
5. View either the *planAhead.log* file or scroll through the Console view to look for any issues encountered during the import process.

Module-level constraints are interpolated into the Project and are included as top-level constraints for implementation.

After successful import, the new placement constraint locations will be displayed in the Device and Package views. Timing constraints can be viewed or modified in the Constraints view. The PlanAhead interface enables further manipulation of the physical or timing constraints.

Creating Multiple Floorplans

Additional floorplans can be created in an existing project. Follow the steps for [“Creating a Floorplan.”](#)

Managing Multiple Floorplans

The Floorplans view provides maintenance commands for the Floorplans in the Project. You can copy, delete, and close Floorplans from the right-click menu in the Floorplans

View. You can also display information about the Floorplan in the Floorplan Properties view. Select **Window > Floorplans** to open the Floorplans view.

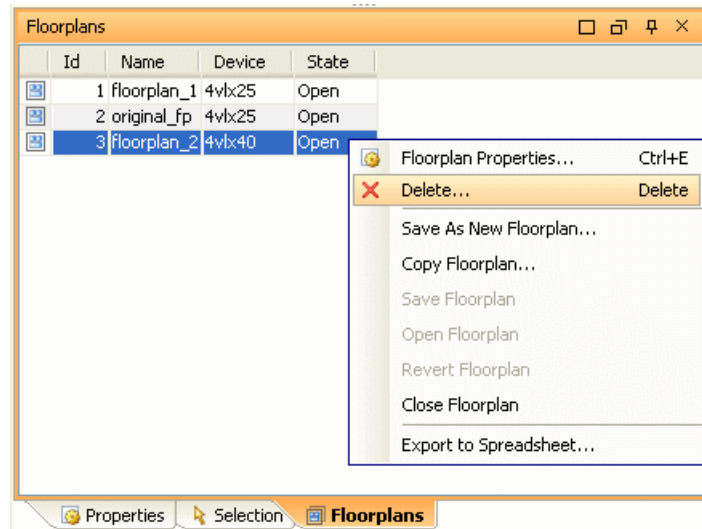


Figure 2-41: Managing Multiple Floorplans

Multiple Floorplans may be created or opened simultaneously to explore design, device, or constraint alternatives. Each Floorplan created references the active imported netlist. The active Floorplan can be toggled by clicking the desired Floorplan tab.

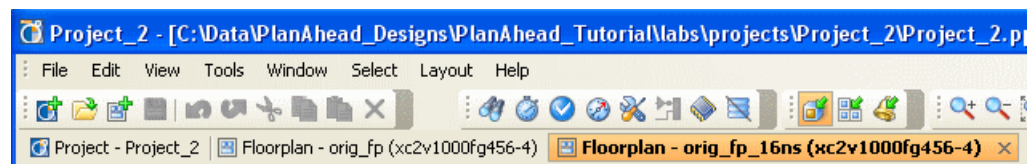


Figure 2-42: Selecting Desired Floorplan to View

Opening multiple Floorplans will use additional system memory so be careful not to open too many.

Viewing or Editing Floorplan Properties

Information can be displayed for any object in the PlanAhead software by selecting an object and examining the information displayed in the Properties view.

To view and edit the properties for a Floorplan:

1. Select the Floorplan (e.g. "fp1_16ns") in the Physical Hierarchy view.
The Floorplan Properties view will appear.

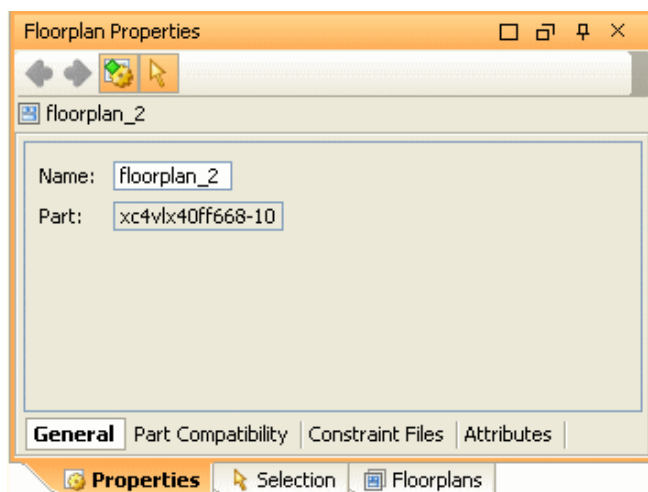


Figure 2-43: Floorplan Properties: General Tab

2. In the Floorplan Properties window, enter a new name in the Name field to identify the Floorplan in this project (e.g. floorplan_2).
3. Click **Apply** to implement any changes entered.
Note: The Cancel button will discard any changes.
4. Click the **Part Compatibility** tab. This tab displays the parts selected to be compatible for I/O pin planning purposes. Parts can be added and removed from the list using the dialog box icon commands. Refer to the [“Defining Alternate Compatible Parts”](#) for more information.

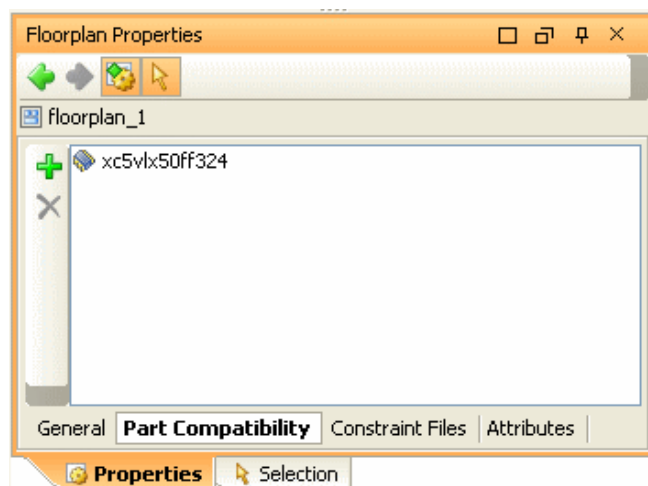


Figure 2-44: Floorplan Properties: Part Compatibility Tab

5. Click the **Constraints Files** tab. This tab displays a list of imported UCF constraint files in the non-editable dialog box.

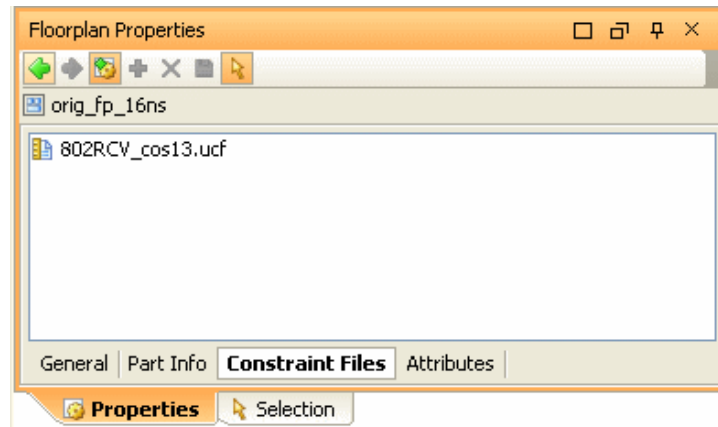


Figure 2-45: Floorplan Properties: Constraint Files Tab

6. Click the **Attributes** tab. In this tab, you can assign attributes to the Floorplan using the **Define new attributes** toolbar button.



Figure 2-46: Define New Attributes Toolbar Button in Properties View

Closing a Floorplan

To close the Floorplan, select **File > Close Floorplan**, or click the **X** icon to the right of the Floorplan tab.



Figure 2-47: Floorplan Tab

A dialog box is displayed to confirm the closing of the Floorplan. Click **OK** to proceed.

If unsaved changes exist in the Floorplan being closed, you are prompted to save the changes. Click **OK** to save the changes.

Closed Floorplans are available within the Project and can be re-opened at any time. All Floorplans that exist in the Project are listed in the Floorplans view.

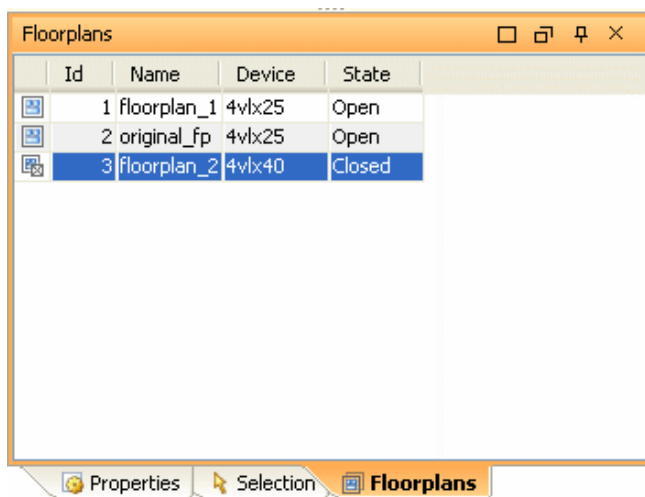


Figure 2-48: Floorplans View

Floorplans can be opened, closed, copied and deleted from the Floorplan view.

Deleting a Floorplan

To remove a floorplan from a Project, and to remove the data from disk, select **File > Delete Floorplan**. The command produces a confirming dialog box, and you are prompted to remove all floorplans and associated runs from disk.

Note: You are prompted to save floorplans if any modifications were made during the PlanAhead session.

Saving a Floorplan

Using Save Floorplan

When a Floorplan is saved, the files in the `Project.data` directory for the saved Floorplan are updated. For more information about which files are saved, see [“Outputs for Project Data.”](#)

To save a Floorplan, select **File > Save Floorplan**, or click the Save Floorplan toolbar button.



Figure 2-49: Save Floorplan Toolbar Button

Using Save As Floorplan

You may elect to modify an existing Floorplan and save it as a new name. This results in the original Floorplan and the newly saved Floorplan both existing in the Project. To rename the Floorplan, see [“Renaming a Floorplan.”](#)

To save a Floorplan to a new name:

1. Select **File > Save Floorplan As**.

The Save Floorplan As dialog box is displayed.

2. In the Save Floorplan As dialog box, enter a new name in the New Name field.
3. Click **OK** to save the Floorplan to the new name.

Any Runs defined in the Floorplan are also copied into the Saved Floorplan. The Runs will have no results data in the Run directories and will display a “Not started” state.

Copying a Floorplan

The primary reason for copying Floorplans is to select a different device or device architecture while preserving the active Floorplan constraints.

To copy a Floorplan:

1. Select the Floorplan tab to make the desired Floorplan active
2. Select **File > Copy Floorplan**.

The Copy Floorplan dialog box displays.

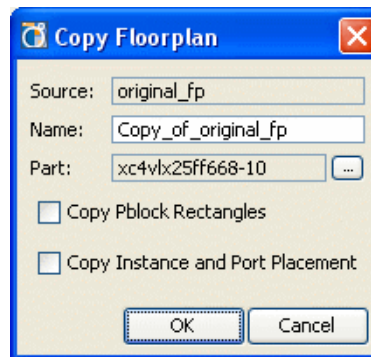


Figure 2-50: Copy Floorplan Dialog Box

3. In the Copy Floorplan dialog box, set the following editable fields:
 - ♦ **Name**—Enter the new Floorplan name. A default name is filled in.
 - ♦ **Part**—Optionally select a new target device.
 - ♦ **Copy Pblocks Rectangles**—Select to copy the Pblock rectangles into the new device using the same tile coordinates, if possible.
 - ♦ **Copy Instance and Port Placement**—Select to copy the placement constraints into the new device using the same coordinates, if possible.
4. Click **OK** to complete the copy.

Renaming a Floorplan

Floorplans can be renamed in the design project. To do so:

1. Select a floorplan in the Physical Hierarchy view, or in the Floorplans view.
The Floorplan Properties dialog box displays in the Properties view.
2. If the Floorplan Properties are not displayed, right-click on the floorplan, and select **Floorplan Properties** from the popup menu.

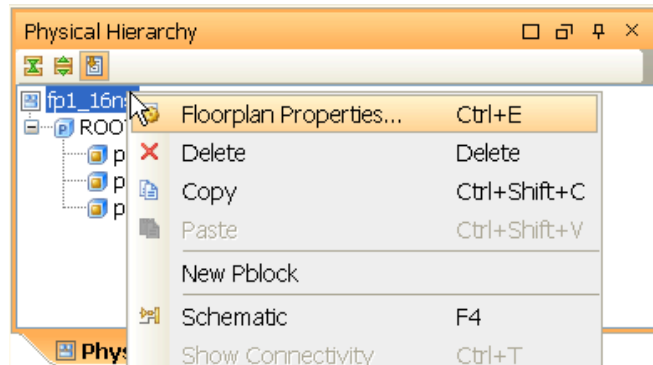


Figure 2-51: Selecting Floorplan Properties

The Floorplan Properties dialog box will display.

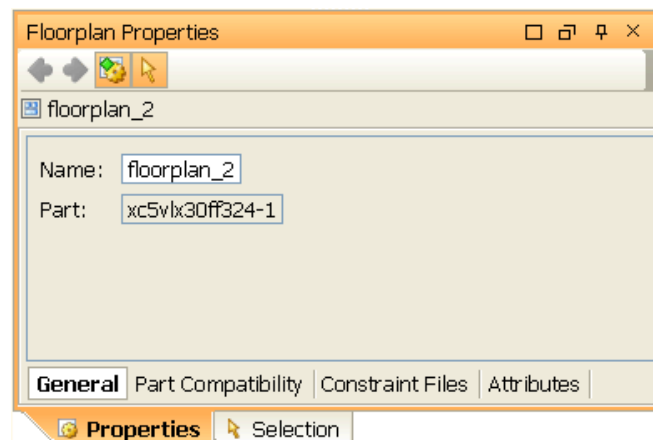


Figure 2-52: Floorplan Properties

3. Enter the new Floorplan name in the Floorplan Properties dialog box under the General tab.
4. Click **Apply** to rename the Floorplan.

Using PlanAhead With Project Navigator

This chapter contains the following sections:

- [“Integration Overview”](#)
- [“ISE and PlanAhead Integration Process”](#)
- [“PlanAhead Viewing Environments”](#)
- [“Transitioning from PACE / Floorplan Editor to PlanAhead”](#)
- [“Transitioning from Floorplanner to PlanAhead”](#)

Integration Overview

The PlanAhead™ software is integrated with the ISE® software in order to perform specific design tasks. When the PlanAhead software is invoked from the ISE Project Navigator environment, PlanAhead is in ISE Integration mode. In this mode, the available PlanAhead features apply only to specific design tasks, including I/O pin planning, floorplanning and timing analysis. A PlanAhead project is automatically created and managed by the Project Navigator environment.

There are four processes in the Project Navigator Processes pane from which PlanAhead can be invoked. They include:

- **I/O Pin Planning (Pre-Synthesis)**
- **I/O Pin Planning (Post-Synthesis)**
- **Floorplan Area/IO/Logic (Post Synthesis)**
- **Analyze Timing/Floorplan Design (Post Implementation)**

The data passed between the two tools and the view layout presented in PlanAhead is dependent on which step is being invoked. Refer to [“ISE and PlanAhead Integration Process”](#) for more information on the mechanics of the integration including data passing and processes.

PlanAhead has two default view layouts for the many design tasks. One is the I/O pin planning environment, also called PinAhead, which contains views pertinent to I/O pin planning and assignment. The other is the PlanAhead Floorplan environment which contains views pertinent to design analysis and floorplanning. It is important to ensure the proper view layout is loaded for the desired design task.

For more information about using the PlanAhead Viewing Environment, see [Chapter 4, “Using the Viewing Environment”](#).

For more information about configuring and loading view layouts, refer to [“Configuring the Viewing Environment”](#).

ISE and PlanAhead Integration Process

Project Navigator and PlanAhead are two independent environments each operating under a separate system process. Steps have been taken to integrate the two processes as much as possible and to ensure data is passed effectively between the two tools. Changes to design data in one tool are not automatically recognized in the other in real time. Users should not attempt to edit logic or constraints simultaneously in both tools. Invoke PlanAhead for the intended purpose and close it before updating the Project Navigator design data. The Project Navigator process steps are synchronized to recognize edits to the UCF file made in PlanAhead once the data is saved. The following sections describe the steps involved and the data transactions that enable the integration.

Passing Logic and Constraints

PlanAhead in ISE Integration mode enables only physical constraint modification for I/O pins, logic LOC and AREA_GROUP constraints. In all cases below, logic connectivity in the form of RTL sources or synthesized netlists are passed to PlanAhead for analysis purposes only. They are not passed back to Project Navigator. All of the PlanAhead features that enable logic or timing constraint modification have been disabled in ISE Integration mode. All logic modification must be performed in Project Navigator, or in an external RTL or synthesis tools. The only files being passed back to Project Navigator are the UCF constraint files. PlanAhead attempts to maintain the original content and format of the UCF files. This includes comments, incomplete constraints, etc. The legality of constraints in the design is not checked upon opening or closing PlanAhead. It is checked during the Translate process step in Project Navigator.

When PlanAhead is invoked, the UCF source files in the Project Navigator project are passed to PlanAhead where physical constraints can be added or modified. When the Save Project command is used in PlanAhead, the modified UCF files are written back to the original Project Navigator source location. If constraint changes are made in PlanAhead and the Exit command is selected, you are prompted to save changes back to the Project Navigator project before the tool closes.

If PlanAhead is invoked and no UCF file exists in the Project Navigator project, you are prompted to create one. This empty UCF file is then passed to PlanAhead.

Project Navigator projects with more than one UCF source file are supported. Before PlanAhead is invoked, a dialog box prompts you to select one of the UCF files. All new constraints defined in PlanAhead will be written to the selected UCF file. Any physical constraints that exist in a non-chosen UCF file will remain in that file, even if the value of that constraint is modified in PlanAhead.

Core-level NCF files used in the Project Navigator design flow are not passed to PlanAhead. To use or view any physical constraints in these files in PlanAhead, you must manually merge them into a top-level UCF file prior to invoking PlanAhead.

A temporary PlanAhead project is created in the ISE project directory, and is removed and replaced each time PlanAhead is invoked from Project Navigator.

I/O Pin Planning (Pre-Synthesis)

You can elect to perform early I/O pin planning prior to having a synthesized netlist. This can be done either by using PlanAhead standalone or by selecting this process step in Project Navigator.

Note: At this stage of the design process, logic synthesis has not yet been run. The tool has no concept of clock ports, clock related logic, differential pairs, GTs, etc. You must be especially careful to ensure these types of ports are placed appropriately to avoid implementation errors. Whenever possible, I/O pin planning should be performed after logic synthesis. The presence of a netlist ensures that the clocks, clock logic, differential pairs, GTs, etc. are recognized and automatically considered during pin assignment in PlanAhead. There are also many more Design Rule Checks (DRCs) that are performed based on logic connectivity and clocks to ensure a legal placement prior to implementation.

To perform I/O pin planning in Project Navigator prior to running synthesis, in the Processes pane, expand **User Constraints** and double-click **IO Pin Planning (PlanAhead) - Pre-Synthesis**, or select the **Tools > PlanAhead > Pre-Synthesis - IO Pin Planning** command.

When PlanAhead is invoked, Project Navigator passes all of the RTL source files, the top-level module name and the UCF file(s) to PlanAhead. PlanAhead is invoked with the default I/O pin planning (PinAhead) view layout displaying. A quick RTL elaboration is performed to extract the top-level I/O ports and displays them in the PlanAhead I/O Ports view.

When the PlanAhead project is saved or closed, the original Project Navigator source UCF file(s) are updated. This will also reset the Project Navigator design process state, if appropriate.

Refer to [“Passing Logic and Constraints”](#) for more information about the integration mechanics and process.

Refer to [Chapter 5, “I/O Pin Planning”](#) for more information about using the PlanAhead I/O pin planning environment (PinAhead).

I/O Pin Planning (Post-Synthesis)

Note: Whenever possible, I/O pin planning should be performed after logic synthesis. The presence of a netlist ensures that the clocks, clock logic, differential pairs, GTs, etc. are recognized and automatically considered during pin assignment in PlanAhead. There are also many more Design Rule Checks (DRCs) that are performed based on logic connectivity and clocks to ensure a legal placement prior to implementation.

To perform I/O pin planning in Project Navigator after running logic synthesis, in the Processes pane, expand **User Constraints** and double-click **IO Pin Planning (PlanAhead) - Post-Synthesis**, or select the **Tools > PlanAhead > Post-Synthesis - IO Pin Planning** command.

When PlanAhead is invoked, Project Navigator passes the synthesized NGC or EDIF format netlist and the UCF file(s) to PlanAhead. PlanAhead is invoked with the default I/O pin planning (PinAhead) view layout displaying. The I/O ports are displayed in the PlanAhead I/O Ports view.

When the PlanAhead project is saved or closed, the original Project Navigator source UCF file(s) are updated. This will also reset the Project Navigator design process state, if appropriate.

Refer to [“Passing Logic and Constraints”](#) for more information about the integration mechanics and process.

Refer to [Chapter 5, “I/O Pin Planning”](#) for more information about using the PlanAhead I/O pin planning environment (PinAhead).

Floorplan Area/IO/Logic (Post Synthesis)

PlanAhead has a robust design analysis and floorplanning environment that can be used prior to or after implementation. To analyze the design or to perform floorplanning from Project Navigator after running logic synthesis and prior to implementation, in the Processes pane, expand **User Constraints**, and select **Floorplan Area/IO/Logic (PlanAhead) - Post-Synthesis**, or select the **Tools > PlanAhead > Post-Synthesis - Floorplan Area/IO/Logic** command.

When PlanAhead is invoked, Project Navigator passes the synthesized NGC or EDIF format netlist and the UCF file(s) to PlanAhead. PlanAhead is invoked with the default PlanAhead design analysis and floorplanning environment displaying.

Note: In Project Navigator, set the Translate process property Macro Search Path (-sd) to the appropriate directory if lower-level NGC format core files are used in the design and are not added as sources.

When the PlanAhead project is saved or closed, the original Project Navigator source UCF file(s) are updated. This will also reset the Project Navigator design process state, if appropriate.

Refer to [“Passing Logic and Constraints”](#) for more information about the integration mechanics and process.

Refer to [“Design Analysis and Floorplanning View Layout”](#) for more information about using the PlanAhead environment prior to implementation.

Refer to [Chapter 9, “Analyzing Implementation Results”](#) and [Chapter 10, “Floorplanning the Design”](#) for more information about using the PlanAhead environment after to implementation.

Analyze Timing/Floorplan Design (Post Implementation)

PlanAhead has a robust design analysis and floorplanning environment that can be used prior to or after implementation. Analyzing the design after implementation allows you to view the placement and timing results in order to comprehend potential design issues. Often, physical LOC or AREA_GROUP floorplanning constraints can help drive the implementation tools toward better and more consistent results, and reduce implementation runtimes.

To analyze the design or to perform floorplanning from Project Navigator after implementation, in the Process pane, expand **Implement Design**, expand **Place & Route**, and double-click **Analyze Timing/Floorplan Design (PlanAhead) - Post-Synthesis**, or select the **Tools > PlanAhead > Post-Implementation - Analyze Timing/Floorplan Design** command.

When PlanAhead is invoked, Project Navigator passes the synthesized NGC or EDIF format netlist, the UCF file(s), the ISE placement data, and timing results to PlanAhead. PlanAhead is invoked with the default PlanAhead design analysis and floorplanning environment displaying. In order for PlanAhead to extract the ISE placement data, a command called **xd1** must first be run. It produces a file with a .xd1 extension and may take a few moments to run. A progress bar is displayed in PlanAhead while this command is running. In order to expedite re-invoking PlanAhead, the interface will first check for the existence of the XDL file and will not regenerate it if it is still current. When you select **Tools > PlanAhead > Post Implementation - Analyze Timing /Floorplan Design** with

the implementation process out of date, you are prompted to either reimplement the design and launch PlanAhead, or launch PlanAhead on the existing result data without rerunning the implementation tools.

When the PlanAhead project is saved or exited, the original Project Navigator source UCF file(s) are updated. This will also reset the Project Navigator design process state, if appropriate.

Refer to [“Design Analysis and Floorplanning View Layout”](#) for more information about the integration mechanics and process.

Refer to the [Chapter 9, “Analyzing Implementation Results”](#) and [Chapter 10, “Floorplanning the Design”](#) for more information about using the PlanAhead environment following implementation.

PlanAhead Viewing Environments

I/O Pin Planning View Layout

The PinAhead environment is invoked automatically when selecting either of the I/O pin planning processes in Project Navigator. It can also be launched from within PlanAhead by selecting **Tools > Open PinAhead**, or **Layout > Load Layout > PinAhead Cockpit**.

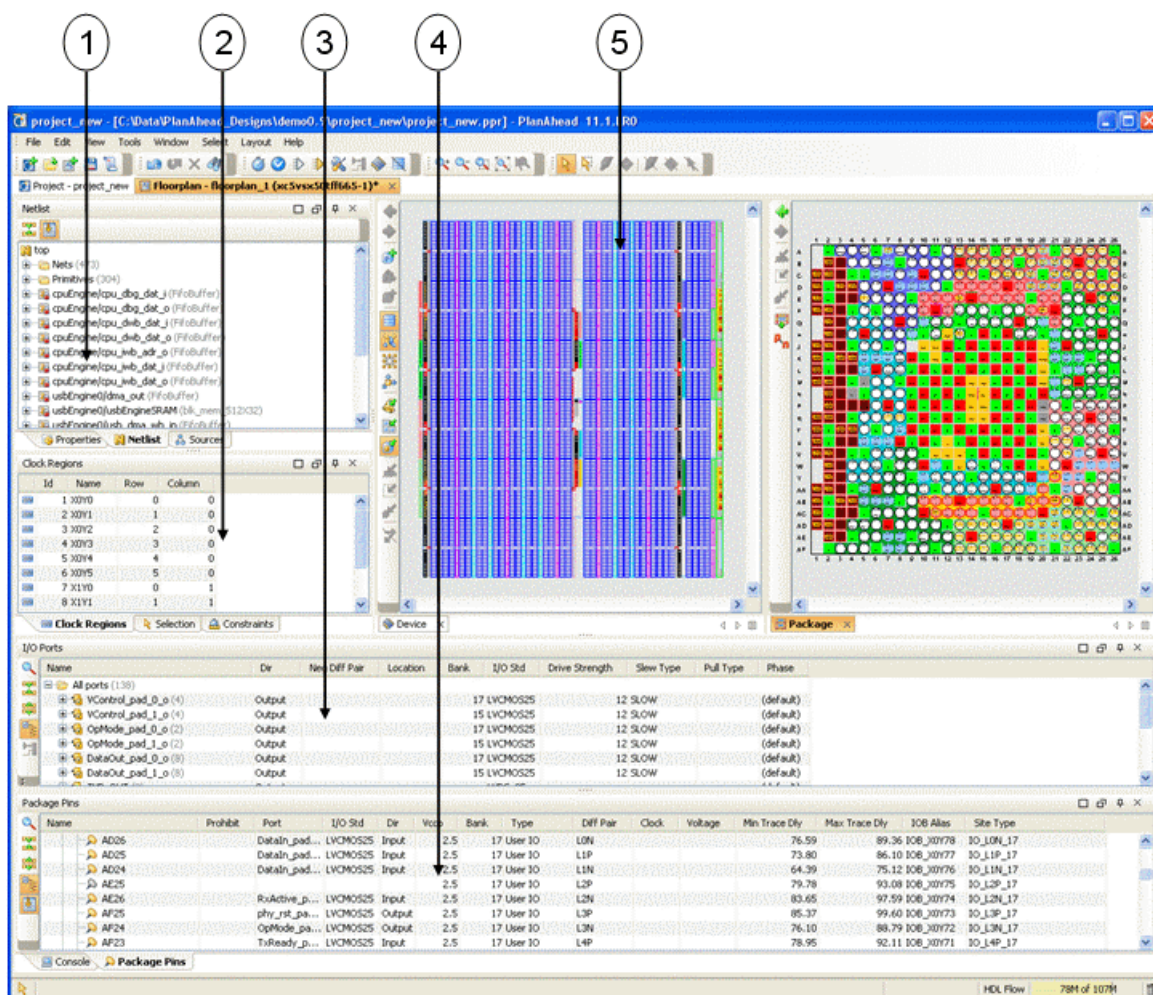


Figure 3-1: PinAhead View Layout in ISE Integration Mode

The areas of the viewing environment are the following:

1. Object Properties, Selected Objects, Netlist, Constraints
2. Clock Regions
3. I/O Ports list
4. Package Pins, Console View, Results Views (Timing, Find, Design Rules Check)
5. Split Workspace—Device View, Package View, Schematic View, Instance Hierarchy View, Reports

The PinAhead environment presents information about the specific device package and the design specific I/O information. Opening the PinAhead environment consists of nothing more than loading an alternate view layout. Use the **Layout > Load Layout > PlanAhead Default** command to return back to the default PlanAhead environment.

For more information about using the PlanAhead Viewing Environment, see [Chapter 4, “Using the Viewing Environment.”](#)

For more information about the I/O pin planning environment, see [Chapter 5, “I/O Pin Planning”](#).

Design Analysis and Floorplanning View Layout

The PlanAhead design and analysis environment is invoked automatically from Project Navigator when you select **Floorplan Area/IO/Logic** or **Analyze Timing/Floorplan Design** from the Processes pane or the **Tools > PlanAhead** menu, as described above. It can also be loaded from within PlanAhead by selecting **Layout > Load Layout > PlanAhead Default**.

When a Floorplan is opened and made active, the default Floorplan environment layout configuration is displayed.

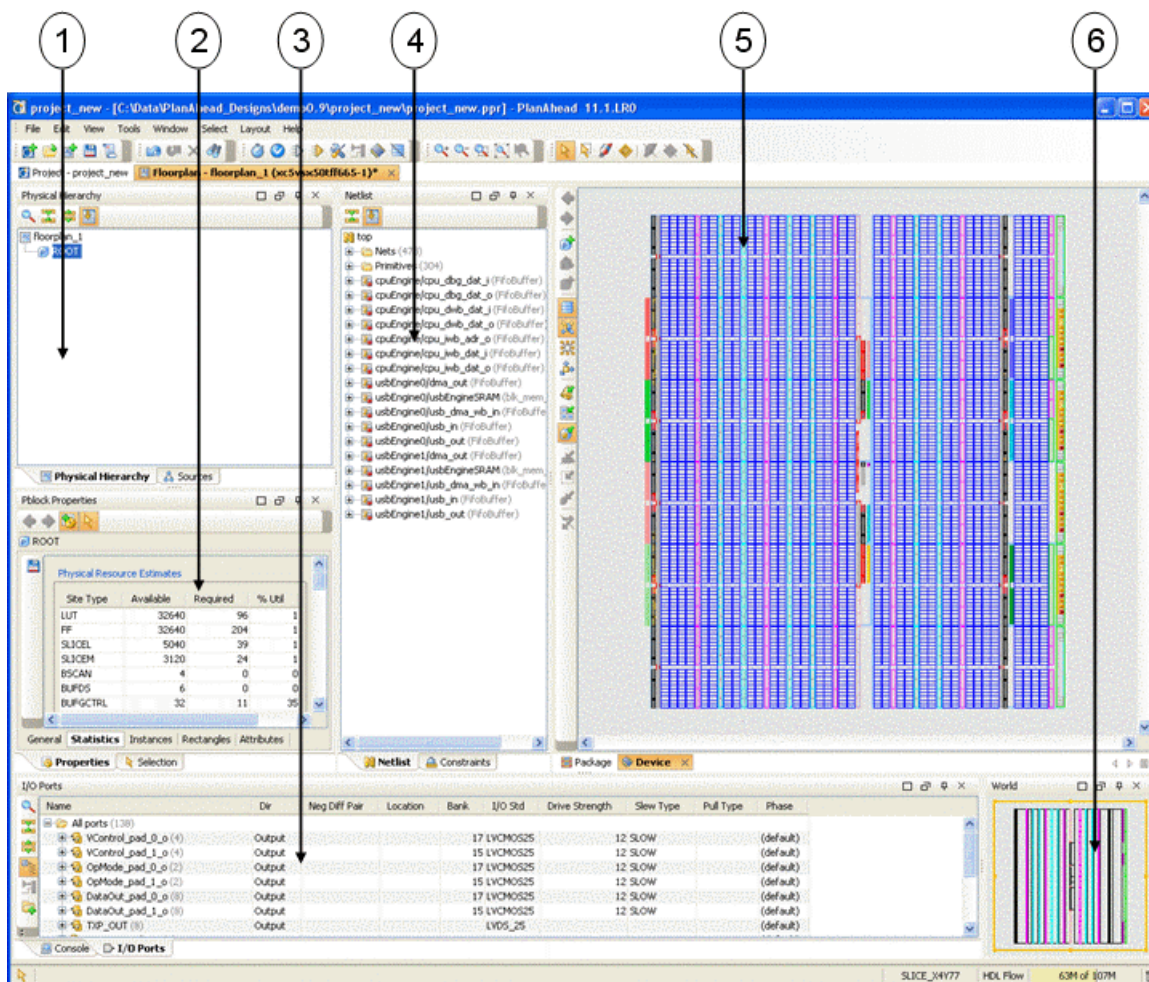


Figure 3-2: PlanAhead Floorplan Environment

The areas of the viewing environment are the following:

1. Physical Hierarchy View, Metrics View, Sources View
2. Object Properties View, Selection View
3. Console View, I/O Ports View, Package Pins View, Results Views (Timing, Find, Design Rule Check, Metrics)
4. Netlist View, Constraints View

5. Workspace—Device View, Package View, Schematic View, Instance Hierarchy View, Reports
6. World View

For more information about using the PlanAhead Viewing Environment, see [Chapter 4, “Using the Viewing Environment”](#).

Transitioning from PACE / Floorplan Editor to PlanAhead

For the ISE Design Suite 11.1, all FPGA pin-planning and floorplanning functionality is provided by PlanAhead. A fully-licensed, and fully-functional version of PlanAhead is supplied as a standard part of the release.

This section explains how to use PlanAhead for common tasks that were previously accomplished with either PACE or Floorplan Editor.

Overview

PlanAhead can be invoked through Project Navigator or directly from the command line prompt. Integrated within the Project Navigator environment, PlanAhead is launched at four different design process steps. They include

- **I/O Pin Planning (Pre-Synthesis)**
- **I/O Pin Planning (Post-Synthesis)**
- **Floorplan Area/IO/Logic (Post Synthesis)**
- **Analyze Timing/Floorplan Design (Post Implementation)**

Each of these steps offers unique and powerful capabilities previously only available in the standalone PlanAhead environment. When PlanAhead is invoked from Project Navigator, the interface provides access only to the PlanAhead features specific to the selected task.

Alternatively, when launched as a standalone PlanAhead environment, you can work with the full range of PlanAhead RTL to bitstream design and analysis capabilities.

For CPLD users, the ISE Design Suite 11.1 still supports the use of PACE for graphical pin-assignment tasks.

This document covers the most common tasks that you would have accomplished using PACE or Floorplan Editor - mainly pin-planning and area-based floorplanning. Each main task will be covered in its own sub-section below, and provide you with a brief description of how to accomplish the same task using the PlanAhead environment.

Launch and Invocation of PlanAhead from Project Navigator

The Processes pane of Project Navigator provides four main processes to accomplish pin-planning and floorplanning tasks: **I/O Pin Planning (Pre-Synthesis)**, **I/O Pin Planning (Post-Synthesis)**, **Floorplan Area/IO/Logic (Post Synthesis)**, and **Analyze Timing/Floorplan Design (Post Implementation)**.

Invoking any of these four processes will launch PlanAhead in a mode to most directly accomplish the selected task. There is full integration between Project Navigator and PlanAhead. This means that Project Navigator will automatically supply all required input files to PlanAhead. Any saved output files from PlanAhead (e.g. UCF based constraints) will be saved to the proper location within the Project Navigator project.

It is important to note that when launched from Project Navigator, PlanAhead is opened in a streamlined "ISE Integration" mode where not all functionality of standalone PlanAhead is available.

Launch and Invocation of Standalone PlanAhead

PlanAhead can be launched by typing **planahead** at the standard command-line shell prompt (Windows or Linux), or on Windows you can double-click on the PlanAhead desktop icon. To start using PlanAhead:

1. Select **Create a New Project** from the launch screen.
2. Follow the New Project wizard.
3. Proceed to pin-planning tasks.

For a full description on how to invoke PlanAhead in a standalone mode, see [Chapter 2, "Creating and Managing Projects."](#)

Depending on what stage you are in the design flow, you can start in the Empty Design, Pre-Synthesis or Post-Synthesis modes. These modes are similar to capabilities that were provided in PACE and Floorplan Editor.

Empty Design & Pre-Synthesis I/O Pin Planning

In "Empty Design" mode, you launch PlanAhead standalone with no pre-existing HDL or netlist files. Select **Do not import sources at this time** in the New Project wizard to open PlanAhead in the "Empty Design" mode.

In "Pre-Synthesis" mode, you have a Verilog or VHDL file that contains (at a minimum) the top-level ports of the design, a CSV spreadsheet, or a UCF constraint file. In PlanAhead, select **File > Import I/O Ports > From HDL/CSV/UCF** to import I/O Ports into an empty project.

In these modes, PlanAhead is typically used as an early pin-planning tool.

Note: Be aware that in the absence of a synthesized netlist, no clock Ports or related clock logic is recognized. You must be very careful to place clock Ports accordingly. When a synthesized netlist is used as input, many more clock, diff pair, GT and clock related logic placement capabilities and design rule checks are available.

Common tasks supported in this Empty Design mode are:

- **CSV Import** - Users often start pin-planning with a simple list of pins that is created in a spreadsheet program like Microsoft Excel, and saved as a CSV (comma separated value) file. PlanAhead can read this file to populate the IO Ports list view by selecting **File > Import I/O Ports**. From there, you can use the drag and drop individual pins (or groups of pins) to the graphical device or package view.
- **UCF Import** - On many designs, you may have a list of pins defined in an existing UCF file, PlanAhead can import this pre-existing UCF file to populate the IO Ports list view. From there, you can use the drag and drop individual pins (or groups of pins) to the graphical device or package view. Similar to importing a CSV file, select **File > Import I/O Ports**. Port direction is not defined in the UCF file format. You should use the Set Direction popup menu command in define Port direction prior to assigning pins.
- **HDL Import** - You may have a top-level HDL file that defines the main IO pins of the design, but is not yet synthesized into an NGC or EDIF netlist. (It may often be the case that this top-level HDL file contains nothing more than the port definitions, so it

may not even be "synthesizable" at this time.) However, just as with PACE or Floorplan Editor, PlanAhead can read in this file and enable you to accomplish early pin-planning and assignment tasks. Just as with CSV or UCF import, a pre-synthesis HDL file can be read in using the Import I/O Ports function of the File menu.

- **Create I/O Ports** - To start pin-planning without any pre-existing input files, you can also interactively create new pin definitions using the Create I/O ports popup menu command in the IO Port view.
- **Export Top-level HDL** - Once pin assignments have been completed, like PACE and Floorplan Editor, you can also export a top-level Verilog or VHDL file with the pin-names populated within the HDL port list. This is a convenience feature to help you start the HDL coding process.
- **Export CSV** - PlanAhead also provides the ability to export a CSV format file of the pin list and information, as this can be used by external applications (e.g. board layout tools).
- **Export UCF** - PlanAhead also provides the ability to export a UCF format constraint file of the pin definitions.

General Notes on Empty Design and Pre-Synthesis pin-planning

- **DRC Checking** - The DRC can be invoked by selecting **Tool > Run DRC**. In the Empty Design or Pre-Synthesis pin-planning modes, there are only limited design rule checks that can be performed on the legality of the pin assignments. These typically include things like IO standard banking rules, but don't include the more sophisticated DRC, such as checking of clocking rules. The more robust DRC set is only enabled when a post-synthesis netlist is available to PlanAhead.
- **Saving Constraints to UCF** - Once pin-assignments are complete, you can save the constraints to a UCF file for further processing by the ISE implementation tools.

The above set of tasks are those that most directly correspond to the early pin-planning tasks as would be done using PACE or Floorplan Editor. For further details on any of the tasks described above, or the broader set of pin-planning capabilities within PlanAhead, see [Chapter 5, "I/O Pin Planning."](#)

Post Synthesis Pin Planning and Area based Floorplanning

In this mode, you have synthesized netlist (EDIF or NGC file) and possibly a pre-existing UCF file. These are the "minimum" set of input files that PlanAhead requires to work in this mode.

When PlanAhead is invoked in the post-synthesis mode, a larger set of features within PlanAhead's environment is available for you to use. For the purposes of this document, the tasks that are most applicable for transitioning from PACE or Floorplan Editor include:

- Pin assignment with the full set of PlanAhead's robust DRCs. This includes clocking rules, IO placement checks, etc. Since this more-complete set of functionality is available in the post-synthesis mode, we recommend working from this mode when possible. DRC can be invoked with the **File > Run DRC**. For a full description of the IO DRCs within PlanAhead, see [Chapter 5, "I/O Pin Planning."](#)
- PlanAhead also has fully-featured set of tools to accomplished area-based floorplanning of your design. Just as with PACE or Floorplan Editor, you can map hierarchical blocks to specific regions of the device using the graphical device view.

- Similar to PACE and Floorplan Editor, you can use PlanAhead to lock down "global" logic, such as place global clock buffers, block RAMs, DSP blocks, PLL/DLLs, etc.
- Differing from PACE and Floorplan Editor, you can also perform detailed "low-level" floorplanning of specific LUTs, FFs, SRLs, etc. using PlanAhead. In this way, PlanAhead is a complete pin-planning and floorplanning tool.

General I/O Pin Planning and Floorplanning Tasks

Transitioning from PACE or Floorplan Editor to PlanAhead is a relatively straightforward process as virtually all of the main operations have direct counterparts in PlanAhead. Below is a list of those general features.

Note: Some of the feature below are not available for Empty Design or Pre-Synthesis modes.

I/O Pin Assignment

- Graphical drag-drop assignment from IO Ports view to the graphical device or package views. In the Package view, both the top and bottom view perspectives are supported.
- Busses can be treated as set of pins and assigned as a whole during the drag-drop operation.
- Similar to pre-defined busses, you can create user-defined groups for group-based pin-assignment.
- In the IO Ports view, you can assign IO properties (e.g. IO standard, drive strength, differential pairing, banks, etc.) to individual pins or groups of pins (busses).
- A Weighted Average Simultaneous Switching Output (WASSO) analysis is available in PlanAhead using **Tools > Run WASSO Analysis**. As an enhancement beyond PACE and Floorplan Editor, the PlanAhead analysis is based on the more complete WASSO rules.
- A Simultaneous Switching Noise (SSN) analysis is available in PlanAhead using **Tools > Run SSN Analysis**.
- Design rule checking (DRC)- To run DRC, select **Tools >Run DRC**.
- For designs with a complete netlist, PlanAhead provides a complete set of IO Banking rule checks.
- Clocking rules are also checked to verify that clock buffer placement is legal and optimal. They are not available in Empty Design or Pre-Synthesis modes.
- Prohibit configuration pins by mode. PACE and Floorplan Editor had a specific feature that enabled you to prohibit the device configuration pins (for example, a JTAG port) from being used by general user IO pins. In PlanAhead, this is a manual process. Select the pins you wish to prohibit in either the Device or Package view, and mark them as prohibited using the Set Prohibit right-click menu command.
- Package migration support. PlanAhead provides a facility to automatically prohibit those pins on footprint-compatible packages within the same device family. In the Package view, use the Make Part Compatible right-click menu command.
- Package-pin flight time display (not applicable to wire-bond packaging). While PlanAhead does not provide a graphical view of package flight time display, trace length values are available in the properties for each IO pin. Select a pin and view the Package Pin view table.

Area Based Floorplanning

- Much like PACE and Floorplan Editor, you can create an area-based floorplan using PlanAhead. In the PlanAhead terminology, this is called creating physical blocks, or PBlocks. Simply select a group of hierarchy from the Netlist view, and use the Draw PBlock right-click menu command or use the Draw PBlock toolbar button. You can now draw a rectangular region in the device view, which will assign the selected block of logic to the drawn site range on the Device view.
- PlanAhead will ensure the drawn regions are capable of fitting the logic in that block. It accounts for logic utilization, carry chain height, and RPM footprints, if applicable.

More information on creating and manipulating area-based PBlocks can be found in [Chapter 10, “Floorplanning the Design.”](#)

Saving the New Constraints in a UCF File

Just as with PACE and Floorplan Editor, the main output of the pin-planning and floorplanning task is a UCF file. Once the project is saved, the UCF file can be used in exactly the same manner as if it had been generated with the previous tools.

Viewing ISE Implementation Results

PlanAhead can import implementation results after running the command line ISE tools. The Create New Project wizard has an option to create a Project from ISE results. The wizard walks you through project creation prompting for the netlist, constraints, placement and timing data. For information on loading results from ISE, [“Creating a Project with ISE Placement and Timing Results.”](#)

Summary

PlanAhead provides most of the functionality you had with PACE and Floorplan Editor, and the transition to using it should be quite straightforward. When launched from Project Navigator, PlanAhead displays only those features that are specific to the requested process, and hides the more fully-featured PlanAhead environment. In contrast, when launched in standalone mode, the full capability of PlanAhead is exposed. While PlanAhead is a very easy tool to learn for pin-planning and floorplanning, it has considerable more analysis, design optimization and design closure capabilities. Refer to the PlanAhead documentation (including the introductory video) for further information - <http://www.xilinx.com/planahead>.

Transitioning from Floorplanner to PlanAhead

For the ISE Design Suite 11.1, all FPGA floorplanning functionality will be provided by PlanAhead. A fully-licensed, and fully-functional version of PlanAhead is supplied as a standard part of the release.

This section explains how to use PlanAhead for common tasks that were previously done using Floorplanner, mainly detailed (LUT, FF, and so forth) based floorplanning and timing path analysis. Each main task will be covered in its own sub-section below, and provide you with a brief description of how to accomplish the same task using the PlanAhead environment. For information about pin-planning or area-based floorplanning, see [“Transitioning from PACE / Floorplan Editor to PlanAhead.”](#)

Launch and Invocation of PlanAhead from Project Navigator

For general, post-implementation floorplanning, PlanAhead can be invoked from Project Navigator using the Floorplan Design/ Analyze Timing (Post-Implementation) process in the Processes pane. Invoking the process launches PlanAhead in a mode to most directly accomplish the selected task. There is full integration between Project Navigator and PlanAhead. This means Project Navigator will automatically supply all required input files to PlanAhead. Any saved output files from PlanAhead (e.g. UCF based constraints) will be saved to the proper location within the Project Navigator project.

It is important to note that when launched from Project Navigator, PlanAhead is opened in a streamlined "ISE Integration" mode where not all functionality of standalone PlanAhead is exposed.

Launch and Invocation of Standalone PlanAhead

PlanAhead can be launched by typing **planahead** at the standard command-line shell prompt (Windows or Linux). On Windows, you can double-click on the PlanAhead desktop icon. To start using PlanAhead:

1. Select **Create a New Project** from the launch screen.
2. Follow the New Project wizard.
3. Proceed to floorplanning tasks.

For a full description on how to invoke PlanAhead in a standalone mode, see [Chapter 2, "Creating and Managing Projects."](#)

For the most common floorplanning tasks, the typical set of input files are the synthesized NGC or EDIF netlist, and UCF file (for saving constraints). If design/timing-analysis tasks are the goal, then a placed NCD file and timing report TWX (or TWR) file are also required. For information on importing ISE results, see ["Creating a Project with ISE Placement and Timing Results."](#)

General Floorplanning Tasks

- Drag/drop objects to the Device view. From the design hierarchy view, any individual primitive logic object (LUT, FF, BRAM, etc.) can be placed by dragging it to the graphical Device view. Select the Create Site Constraint Mode toolbar icon in the Device view prior to dragging primitive logic onto Device sites. This results in a LOC constraint in the UCF which directs the placer to place that object at the specified site. Any object type can be floorplanned in this manner.
Note: PlanAhead refers to user constrained LOCs as "fixed".
- Pin-placement and area-based floorplanning functionality is integral to PlanAhead, so full DRC checking is enabled for these types of operations.
- PlanAhead does not provide options to rotate, mirror or flip composite objects during placement. While this was more useful in older device architectures, we no longer recommend using these techniques with the current device families.
- RPM creation. Using the Export-IP command, you can export a block of hierarchy as a separate netlist. If objects within that block also have placement constraints, those constraints can also be exported. This is analogous to Floorplanner's RPM creation capability. The main limitation here is that this is limited to pre-defined blocks of design hierarchy. Export-IP does not support an arbitrarily (non-hierarchical) associated set of logic.

- Pre-existing RPMs are better identified within PlanAhead than they were in Floorplanner. Existing RPMs are listed in the Physical Hierarchy view and can be floorplanned or examined using the Pblock Properties view statistics.
- BEL-level constraints can also be written out directing objects to specific sites within a slice (e.g. which particular FF within the slice). There are two modes within PlanAhead that control this behavior - **Create Site Constraint Mode** and **Create BEL Constraint Mode**. If a object's placement is read in from a placed NCD file, unlike with Floorplanner, BEL locations are preserved by default.
- Re-creating design hierarchy from a flat netlist. Floorplanner had a "Group by Hierarchy" function to recreate a name-based hierarchy. While not 100% accurate, it was useful for design analysis and area-based floorplanning. PlanAhead does not provide this functionality, and we recommend setting the appropriate control in the synthesis tool to either keep or rebuild the design hierarchy during synthesis. This is a much more reliable and robust method to visualize the true hierarchy of the design. Use the XST option which creates hierarchical netlists (-netlist_hierarchy rebuilt) to accomplish this.
- Floorplanner color coded the hierarchy by default, which is a manual process in PlanAhead. You can add color to the hierarchy by selecting a hierarchy block in the Netlist view, selecting **Highlight Primitives** from the context menu, and then selecting **Cycle Colors** from the context menu. The colors are not stored with the design.

Timing Analysis - Path Visualization

Floorplanner could be used in conjunction with Timing Analyzer to view the placement of individual timing paths. To do this, you would select a path in the timing report and direct Timing Analyzer to "cross-probe" the path to Floorplanner. Floorplanner would then highlight the objects in that timing path on the device view and show the connections of the path. Often, a timing problem could be traced to some sort of placement issue.

PlanAhead supports reading in and displaying information from the placed NCD file and the post-route timing report (TWX/TWR), which makes it quite easy accomplish the same task within the PlanAhead environment. When launched from Project Navigator, these files will be passed to PlanAhead automatically (if they are present in the project). PlanAhead does not currently support being a cross-probe "target" from ISE Timing Analyzer.

Timing paths are listed in the Timing Results tab within PlanAhead. Selecting individual paths will cause them to be highlighted in the device view. PlanAhead goes beyond what Floorplanner could provide in that you can also cross-probe a path or group of paths to a schematic-based view. From the Timing Results tab, select the path(s) to display and select the Schematic option in the right-click menu (or press the F4 key.) By providing both physical device (floorplan) and schematic representations of the timing path, all within the same environment, you are provided a rich set of views and tools with which you can analyze the timing of your design. Should you choose to use floorplanning constraints as a means to help improve timing performance, the functionality is completely integrated within the same PlanAhead environment. For more information, see [Chapter 9, "Analyzing Implementation Results."](#)

Viewing ISE Implementation Results

PlanAhead can import implementation results after running the command line ISE tools. The Create New Project wizard has an option to create a Project from ISE results. The wizard walks you through project creation prompting for the netlist, constraints, placement and timing data. For information on loading results from ISE, “[Creating a Project with ISE Placement and Timing Results.](#)”

Summary

PlanAhead provides all of the main functionality for common tasks previously done in Floorplanner, and the transition to using it should be quite straightforward. When launched from Project Navigator, PlanAhead presents only those features that are specific to the requested process, and hides the full-featured PlanAhead environment. In contrast, when launched in standalone mode, the full capability of PlanAhead is exposed. While PlanAhead is a very easy tool to learn for basic floorplanning tasks, it has considerably more analysis, design optimization and design closure capabilities. Refer to the PlanAhead documentation (including the introductory video) for further information - <http://www.xilinx.com/planahead>.

Using the Viewing Environment

This chapter contains the following sections:

- [“The Viewing Environment”](#)
- [“Navigating Views”](#)
- [“Using the Workspace Views”](#)
- [“Using Common Environment Views”](#)
- [“Understanding Object Selection Options”](#)
- [“Configuring the Viewing Environment”](#)

The Viewing Environment

The PlanAhead™ software has a dynamic viewing environment that consists of various view layouts that present the pertinent design and device information for the design task at hand.

PlanAhead can be used as a standalone software tool, or launched for specific purposes from the ISE® software. The full suite of PlanAhead features is available when launched as a standalone, while only a subset of specific features are available when launched from Project Navigator. Refer to [Chapter 3, “Using PlanAhead With Project Navigator”](#) for more information about integration with Project Navigator.

PlanAhead can be used to control each major step of the FPGA design process, including RTL development and analysis, logic synthesis, physical design analysis, floorplanning and implementation control with the ISE software.

PlanAhead enables you to define view layout configurations in order to customize the types of views displayed and their appearance. The tool provides default view layouts for I/O pin planning, RTL development and analysis, and design analysis and floorplanning. These views are called the PinAhead environment, the Project environment and the Floorplan environment, respectively.

PlanAhead enables you to create several different types of Projects that vary depending on the input formats. The input formats include RTL source files, synthesized netlists, I/O ports lists, and implementation results. The view layout being displayed is determined by the type of Project being created. For more information on the PlanAhead project types, see [Chapter 2, “Creating and Managing Projects.”](#)

You can select the tabs just below the toolbar buttons to see the Project or Floorplan environment.

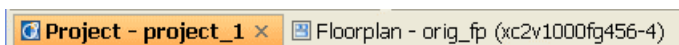


Figure 4-1: Project Environment is Selected

The PinAhead environment can be invoked by selecting the **Tools > Open PinAhead** command.

PinAhead Environment View Layout

I/O Pin planning capabilities are included in PlanAhead to enable I/O and pin assignment and clock planning. The PinAhead environment, which is invoked from any active Floorplan, displays information specifically for I/O pin planning purposes. For more information on the I/O pin planning view, see [Chapter 5, “I/O Pin Planning”](#).

Select **Tools > Open PinAhead** to display the PinAhead environment layout.

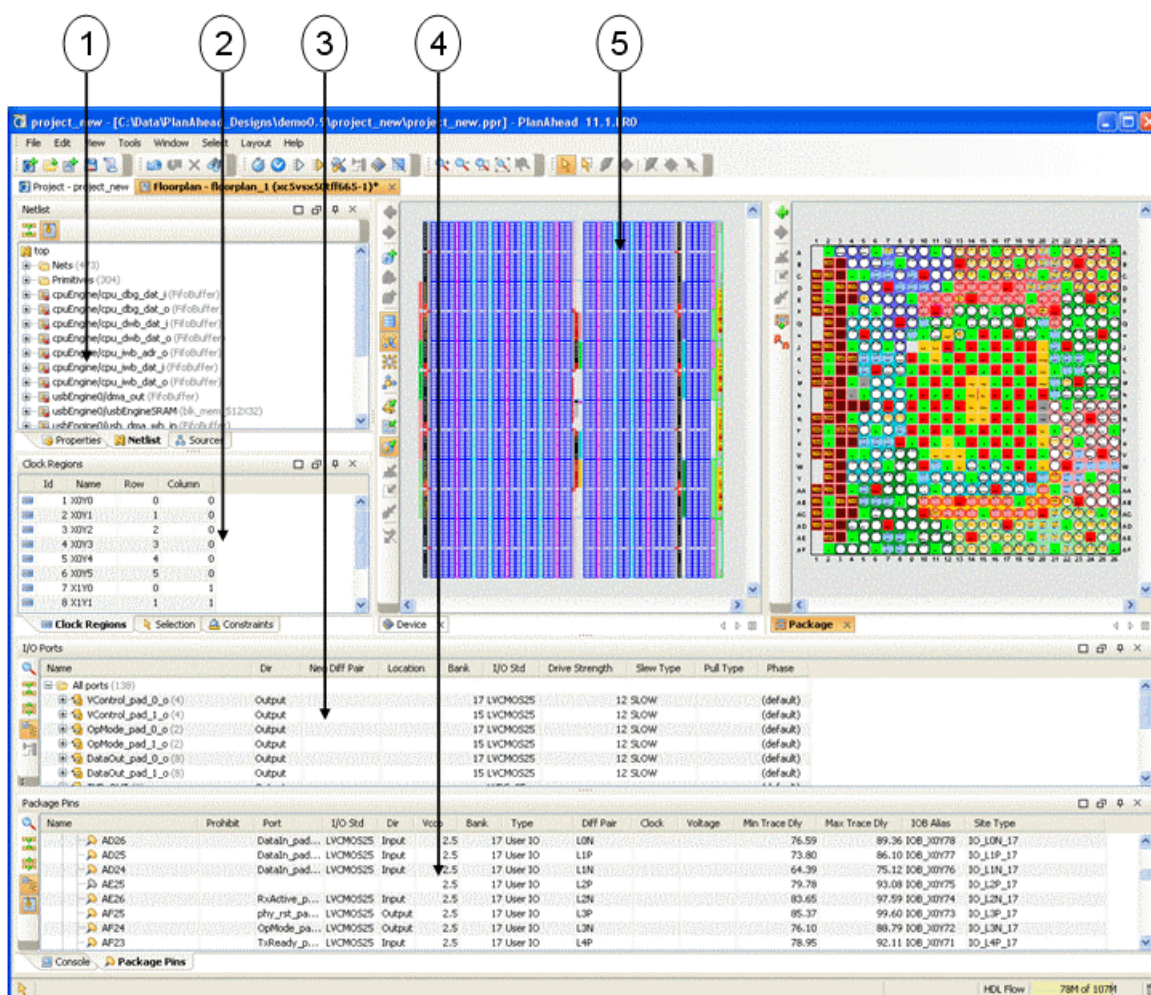


Figure 4-2: Default PinAhead View Layout

The areas of the viewing environment are the following:

1. Object Properties, Netlist, Sources, Physical Hierarchy
2. Clock Regions, Selected Objects, Constraints
3. I/O Ports list
4. Package Pins, Console View, Results Views (Timing, Find, Design Rules Check, SSN)

5. Split Workspace—Device View, Package View, Schematic View, Instance Hierarchy View, Reports, WASSO Results

The PinAhead environment displays information about the specific device package and the design specific I/O information. To open the PinAhead environment, simply load this alternate view layout. Then, select **Layout > Load Layout > PlanAhead Default** to return back to the default PlanAhead environment.

Project Environment View Layout

A Project environment is available for RTL development and analysis, launching synthesis runs, launching implementation runs, and managing the overall Project. For more information, see [Chapter 2, “Creating and Managing Projects.”](#)

When an RTL based Project is opened or when the Project environment tab is selected in below the toolbar buttons in the PlanAhead window, the Project environment layout is displayed. The views that display reflect the state of the design.

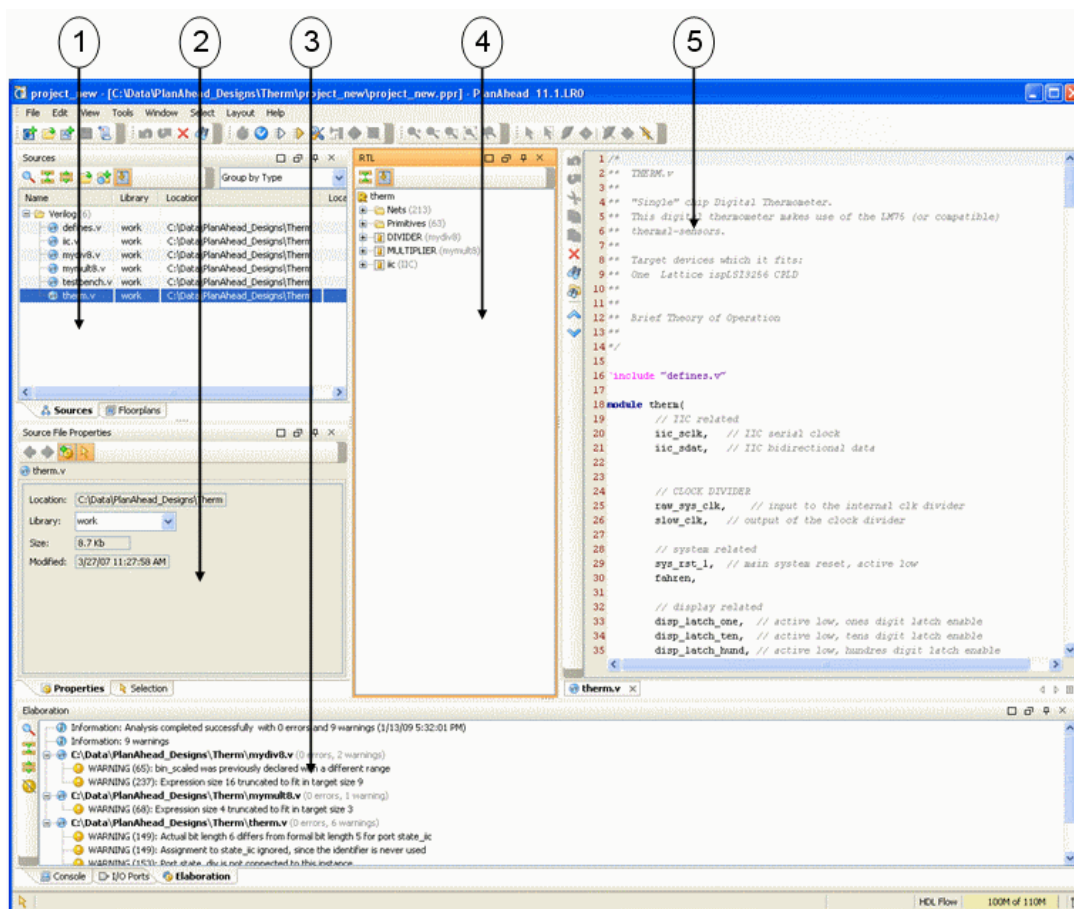


Figure 4-3: PlanAhead Project Environment

For the Project environment, the areas of the viewing environment are defined as follows:

1. Floorplans View, RTL Sources View
2. Object Properties View, Selection View

3. Console View, RTL Elaboration Results View, Design Runs View, I/O Ports View, DRC Results, Find Results
4. RTL Netlist View
5. Workspace— RTL Editor, Schematic View, Hierarchy View, Reports

Floorplan Environment View Layout

When a netlist-based project is open, an empty Project is open, or a Floorplan is active, the default Floorplan environment layout configuration is displayed. For more information about using the views in the Floorplan environment, see [Chapter 10, “Floorplaning the Design.”](#)

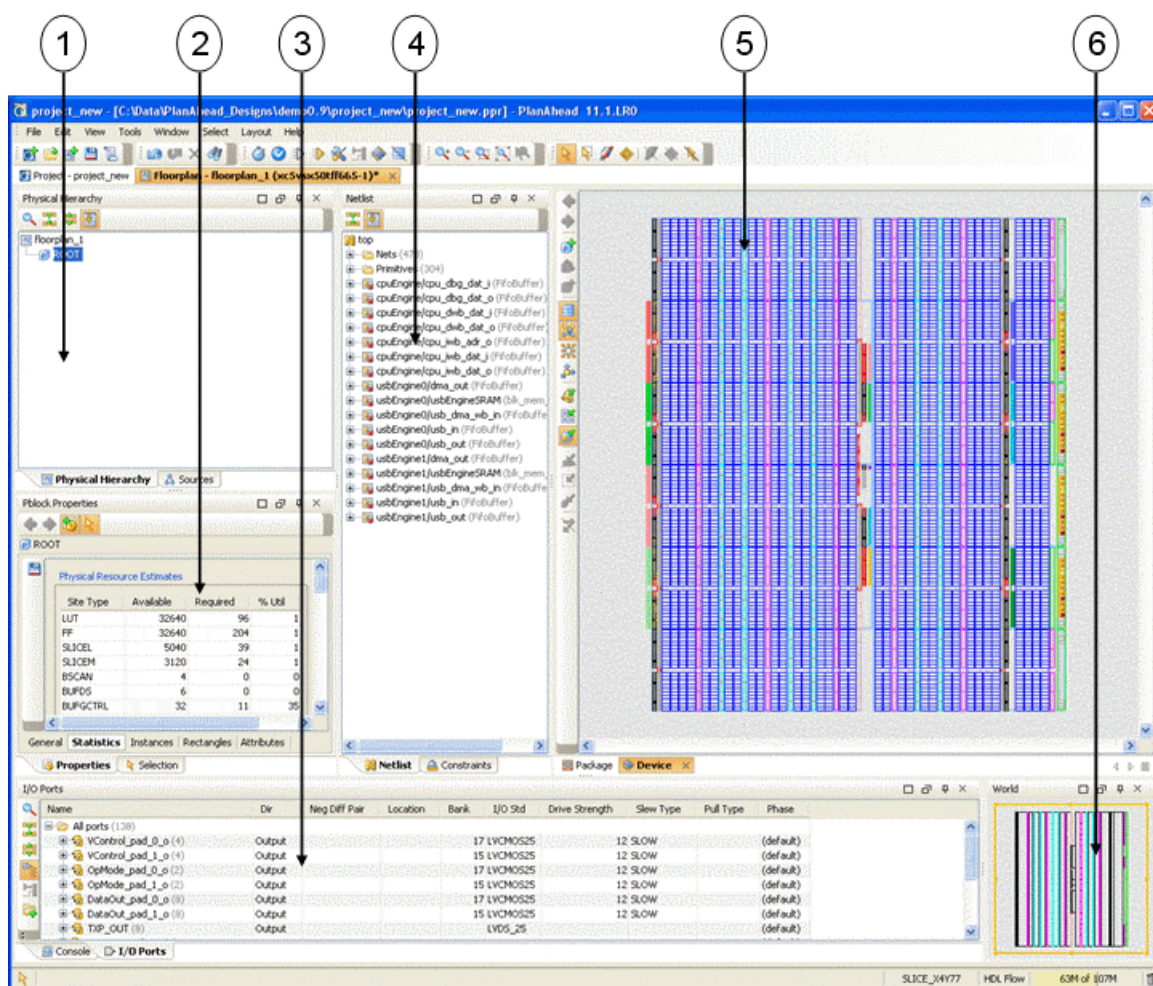


Figure 4-4: PlanAhead Floorplan Environment

The areas of the viewing environment are the following:

1. Physical Hierarchy View, Metrics View, RTL Sources View
2. Object Properties View, Selection View, Floorplans View
3. Console View, I/O Ports View, Package Pins View, Design Runs View, Results Views (Timing, Find, Design Rule Check, Metrics)

4. Netlist View, Constraints View
5. Workspace—Device View, Package View, Schematic View, Instance Hierarchy View, Reports, WASSO Results
6. World View

Navigating Views

Toggling Views Displayed

Tabs are created in the viewing areas for each view. Views are activated by clicking on the tabs. Multiple tabs are allowed for certain view types. This may take a moment to get used to so we suggest you take a moment to experiment with the view manipulation as described below.

Selecting objects in one view will result in the same objects being selected in all other appropriate views. This cross-selection capability makes logic exploration and floorplanning much easier. The default layout attempts to maximize the information displayed when various objects are selected.

The overall size of these viewing areas can be stretched by sliding the view borders. The cursor will change to a slider symbol allowing view border stretching.

The active view displayed in any viewing area is toggled by using the tab interface at the bottom of the view area. Selecting the view tab will make it active.



Figure 4-5: Netlist View Tab is Selected

Double-click the view tab to display the view in the full screen. Workspace views can be restored by double-clicking the view tab again. Non-Workspace views can be restored by selecting the **Maximize/restore** icon in the view banner. For more information, see [“Manipulating Views using the View Banner Commands.”](#)

The viewing areas can be split either horizontally or vertically to display multiple views at once. Each split area can dock multiple views. The possible configuration options are quite extensive.

Each view has a control box to manipulate the view. The views can be floated, hidden, maximized or closed. Closing the view or viewing area will force the other views to resize and claim the space. Control box commands are covered later in this chapter.

The Workspace behaves differently than the rest of the view docking areas. The view types that can be assigned to it are fixed. They include the Device, Schematic, Hierarchy and Package views, as well as the timing path reports, RTL Editor, WASSO analysis results, ISE software reports and the Getting Started page. Multiple views of each view type can be opened and interacted with simultaneously in the Workspace area. The Workspace may be split any number of times either horizontally or vertically. Views may be closed by selecting the Close button in the view. The Workspace view can be maximized or floated by selecting the view tab

Opening Views

Select the **Window** menu from the Main Menu to provide commands to open most window types. Select a window that is already open to make it the active window. For a list

of Window menu commands and a brief description, see “[Toolbar Commands](#)” in [Appendix A](#).

The Schematic view requires at least one object to be selected and is opened using the popup menu **Schematic** command or the **Schematic** toolbar button.





The Properties view requires at least one object to be selected and is opened using the popup menu **<Object type> Properties** command.

Select **New Device View** or **New Package View** to open an additional view in the Workspace.

Manipulating Views using the View Banner Commands

Each viewing area can be manipulated using common window environment commands.

Table 4-1: View Banner Commands

Toolbar Button	Command	Description
	Maximize/restore	Maximizes and restores non-Workspace views.
	Toggle floating	Floats and docks non-Workspace views. Views may be floated outside of the PlanAhead Desktop.
	Toggle auto-hide	Hides and restores the view. When hidden, the entire Docking area appears as a tab at the perimeter of the PlanAhead Desktop. The other docking areas will occupy the space. To restore the view, click the icon displayed on the tab.
	Close	Closes the window, leaving other tabs or docking areas to occupy the space. To re-opened a window or tab, use the Window menu commands.

Each of these commands is also available using the right-click popup menu in the banner of the view.

Using the View Auto-Hide Capability

The viewing areas can be placed in Auto-hide mode to help control the view display. To initiate auto-hide mode, click the view banner **Toggle auto-hide** button, or select the **Auto-hide** command from the popup menu.

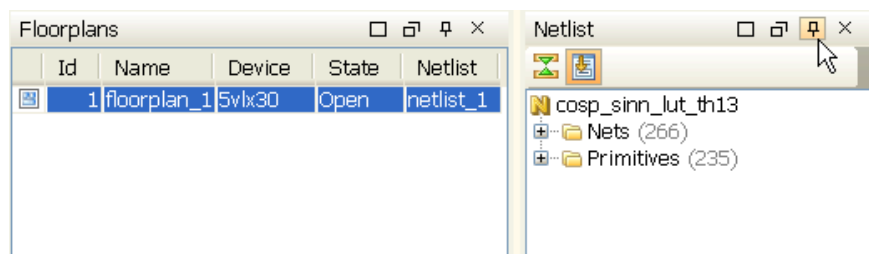


Figure 4-6: Click the Toggle Auto-hide Button in the Netlist View

The entire Docking area will appear as a tab on the perimeter of the PlanAhead Desktop. The other docking areas will grow to occupy the space.

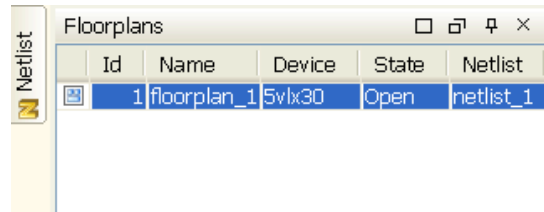


Figure 4-7: Netlist View in Auto-hide Mode

Each auto-hide tab has an icon that can be used to restore the view to its original location. Hover the cursor over the appropriate auto-hidden tab to display the view temporarily. To re-dock an auto-hidden view, click the **Toggle auto-hide** button, or select the **Auto-hide** command from the popup menu once again.

To undo the auto-hide mode, select **Layout > Undo Toggle Autohidden/Docking Mode**.

Floating Views

Views can be undocked or “floated” so that they may be moved and sized independently. To float a window, click on the view tab or banner, and or click the **Toggle floating** button or select **Floating** from the popup menu.

The view will appear in a separate floating window. In this case, windows will obviously overlap.

Floating windows can be moved by dragging the view banner. They may also be moved outside of the PlanAhead main window.

The default locations and sizes to display all floating view types are also stored in your saved layouts.

The Workspace view is floated by selecting the view tab and then selecting the **Float Window** popup menu command.

Defining Viewing Area Sizes

The window borders between the main docking areas can be dragged to resize any of the PlanAhead docking areas views. The cursor will change to a slider bar indicating that the view border is selected for dragging. The viewing areas can also be removed or hidden to provide more viewing space for other views. If a view is closed, the other docking areas will fill in the void.

Using the View Specific Toolbar Commands

Many views have toolbar commands that are displayed within the view. The most common View-specific toolbar commands are described below.

Table 4-2: Common View-Specific Toolbar Commands







Toolbar Button	Command	Description
	Collapse All	Collapses the entire expanded tree as in the Netlist, Constraint and Physical Hierarchy views.
	Expand All	Expands the entire expanded tree as in the Constraint and Physical Hierarchy views.

Table 4-2: Common View-Specific Toolbar Commands

Toolbar Button	Command	Description
	Automatically scroll to selected objects	Toggles the particular view to automatically expand and scroll the tree to expose newly selected objects, or whether to remain static. The default in all windows is to automatically expand and scroll.
	Automatically update the contents of this window	Toggles the Properties view to automatically update to display properties for newly selected objects in other views, or to remain static on the selected object. The default is to automatically update.
	Previous Object	Navigates backwards through the list of previously selected objects. In the Workspace Views, cycles back through the various Zoom levels that were displayed.
	Next Object	Navigates forward through the list of selected objects. This button is enabled after you backtrack using the Previous Object button. In the Workspace Views, cycles forwards through the Zoom levels that were displayed.

There are view banner Toolbar buttons not described here. For information on the View Banner commands, see [“Manipulating Views using the View Banner Commands.”](#)

Using the Workspace Views

What are the Workspace Views?

The PlanAhead Workspace is the graphic viewing area, which displays the graphical views and some report and log information.

The views displayed in the Workspace are the following:

- RTL Editor
- Device
- Package
- Schematic
- Instance Hierarchy
- Getting Started jump page

- Reports and log files
- WASSO Results

Opening Workspace Views

The Device and Package views may be opened in the Workspace by using the **Windows** Main Menu commands. Multiple views of the same view type may be opened within the Workspace area. For example, two Device views can display different areas of the device. To open a new Device or Package view, select **Window > New Device View** or **Window > New Package View**.

The Schematic views are opened a little differently. To open a Schematic view:

1. Select at least one object to display in schematic format.
2. Select the **Schematic** command from the popup menu, press **F4**, or click the **Schematic** toolbar button.



Figure 4-8: Schematic Toolbar Button

The Schematic displays in the Workspace. Running subsequent Schematic commands will open additional Schematic views in the Workspace.

The Getting Started page can be displayed by selecting **Help > Getting Started**.

Viewing the Workspace Full Screen

Double-clicking the Workspace tab will result in the view being displayed full screen. Workspace views can be restored by double-clicking the view tab again.

You can also right-click on the tab to view the Workspace tab popup menu. Select either **Maximize Workspace** or **Restore Workspace** from the popup menu depending on the state of the Workspace.

Floating the Workspace View

To Float the Workspace view, select the Workspace view tab and use the **Float Window** popup menu command.

Printing the Workspace View

You can print the View in focus in the Workspace--Device View, Package View, Schematic View, Instance Hierarchy View. Select **File > Print** to print the current viewable area.

Closing Workspace Views

The views within the Workspace can be closed by clicking on the **X** icon in a view tab.



Figure 4-9: Device View Tab with X Icon

Splitting the Workspace

The Workspace viewing area can be split horizontally or vertically to enable multiple views to be displayed simultaneously. To do so, use the popup menu in the Device view to select the desired split.

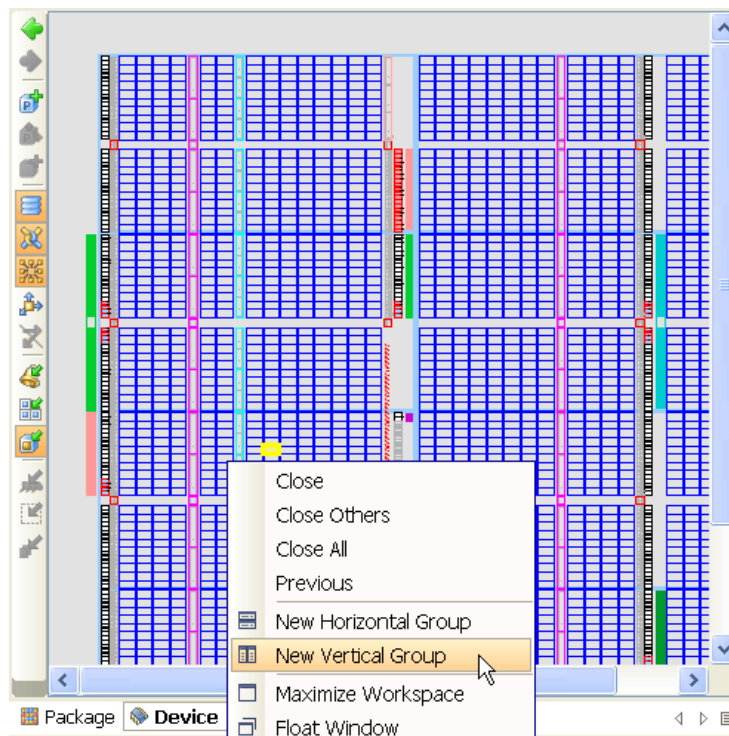


Figure 4-10: Splitting the Workspace Vertically

Each panel now acts as an independent Device view allowing multiple views to be docked for viewing.

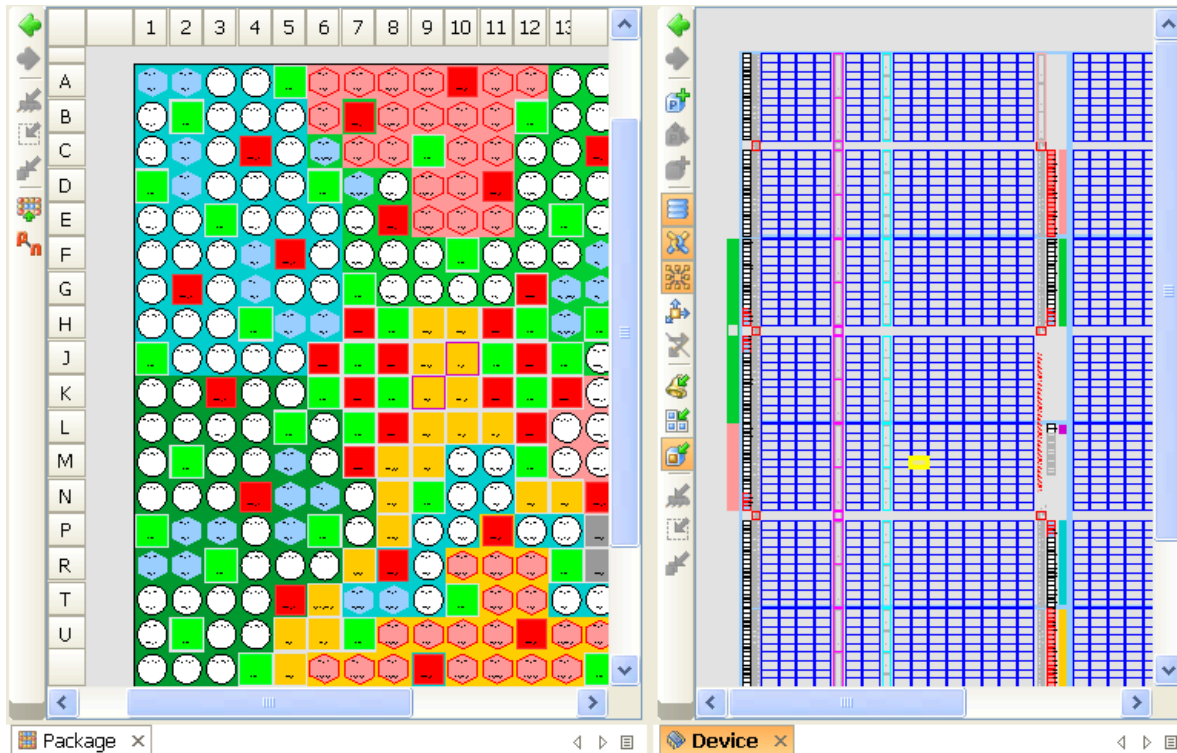


Figure 4-11: Two Workspace Views Displayed Vertically

Multiple views of the same type can be opened, such as two Device views for viewing different areas of the device or different zoom levels.

Using Common Environment Views

Using the Console View and Tcl Command Line

The Console view displays messages from previously executed Tcl commands. All messages are also written to the `planAhead.log` file. Indications of command errors, warnings and successful completion are echoed to this window. To invoke the Console View, select **Window > Console**.

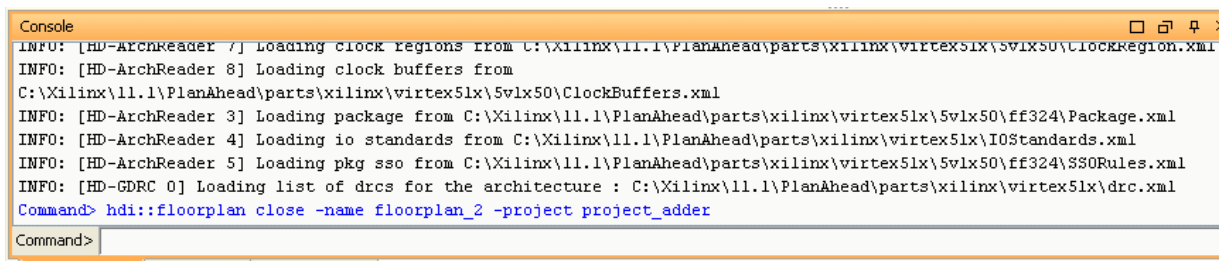


Figure 4-12: Console View

Using the Tcl Command Line

The Command Line (shown above) enables manual command entry using Tcl format PlanAhead commands. Commands are entered by clicking on the command line and

typing them in the Command dialog box entry. Every editing command that can be performed using the menu or direct manipulation (e.g. drag & drop) has an equivalent Tcl command. When invoking a command in the interface (i.e. menu or direct manipulation), the equivalent Tcl command is invoked and displayed on the message area, and written to the `planAhead.jou` file.

Command history can be accessed using the **Up arrow** and **Down arrow** keys in the Command Line window.

Using Tcl Help

Command line help is available for all commands by using the following syntax at the command line:

```
Command> hdi::?
```

or

```
Command> help
```

More detailed information about the commands can be retrieved by extending the help query.

```
Command> hdi::pblock ?
```

For explicit command syntax, perform the command once, and then view the `planAhead.jou` file in the PlanAhead invocation directory.

Using the World View

The World view, located at the bottom right corner, displays a less detailed overview of the active Workspace view. To invoke the World View, if hidden, select **Window > World**.

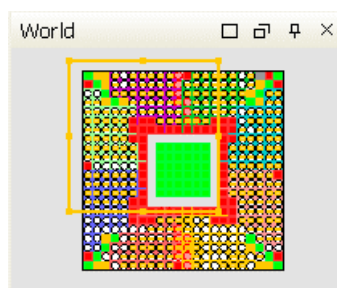


Figure 4-13: **World View**

The World View will reflect the zoom area and selected objects for the active view for the Schematic, Device, Package and Hierarchy Views.

A navigation rectangle displays the area that is visible in the active view. The navigation rectangle can be dragged to reposition the display area in the active view. The navigation rectangle can also be resized by dragging one of the resize handlers. This adjusts the active view scale factor (i.e. zoom in or zoom out) to match the new display area defined by the navigation rectangle.

Selected Pblocks, instances and I/O Ports are highlighted in the World view for easier location.

Using the Status Bar

The status bar at bottom of the PlanAhead Desktop displays useful information.



Figure 4-14: Status Bar

The parts of the status bar are defined and described below.

Status Mode Field

The left side of the status bar shows an icon representing the current mode. The default mode is the “select” mode, indicated by a yellow arrow cursor. Other modes include zoom area, draw Pblock, and an hourglass to indicate that PlanAhead is waiting for your input.

Information Message Field

The second field in the status bar is used to display context-sensitive information. For example, when the cursor is in the device view or the schematic view, this field contains the name of the instance directly under the cursor. The information message field also contains detailed description of PlanAhead commands when the mouse is over its corresponding toolbar button or menu item.

Coordinates Field

To the right of the Information Message Field is the Coordinates Field. As the cursor is moved over BRAM, DSP48, etc. in the Device view, this field displays the name and coordinates. If the cursor is over a pin in the Package View, this field displays pin information, such as coordinates, type and name.

Mode

The Mode indicates what type of Project or process step is being used. When PlanAhead is invoked from ISE Project Navigator, the ISE Integration mode is displayed.

Java Memory Consumption Gauge

The PlanAhead graphical user interface has a 512MB memory limit on Windows, or 1GB if the tool is invoked on 64-bit Linux. There is a memory gauge in the bottom left corner that shows how much of this memory is utilized. If the gauge goes within 10% of the limit you should save work, and restart the tool. This gauge only shows Java memory consumption. Task Manager or Top reports Java and C memory consumption, and displays a higher value.

Understanding Object Selection Options

PlanAhead enables you to select, highlight and mark objects.

Selecting Objects

Selecting an Object

Objects can be selected in many ways in PlanAhead. Click the object to select it in the current view. When selected in any view, objects also become selected in the other appropriate views.

To move objects, hold the left mouse down and drag it, release to drop it on the desired location. The cursor will change to a hand symbol when the move mode is activated.

When objects overlap a priority scheme is used where the smaller size objects are selected. If objects become difficult to select in the Device view, use the Physical Hierarchy or Netlist views to select them. Objects can always be selected from either of these two views regardless of the Selection Rule setting in the PlanAhead Options dialog box.

If you experience difficulty selecting the correct object, the Select commands in the popup menu can be used to select a specific item within the stack of items under the cursor.

Selecting Multiple Objects

In most selectable views and dialog box lists, multiple objects may be selected by pressing and holding the **Shift** key to select a range of elements in a tree or table. The **Ctrl** key is used to select multiple elements individually.

Using the Select Area Command

You can draw a rectangle in any of the Workspace views in order to select objects. To do so:

1. Choose **Select > Select Area**, or click the **Select Area** toolbar button.



Figure 4-15: **Select Area Toolbar Button**

All objects that the rectangle surrounds or touches are listed in the Select Area dialog box, which enables you to filter selection by type.

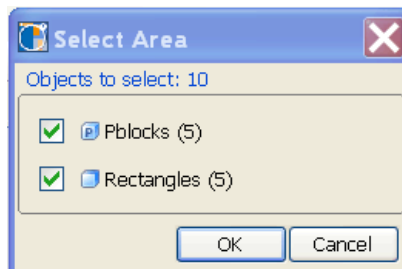


Figure 4-16: **Select Area Dialog Box**

2. Object types can be filtered from being selected by turning off the check box.
3. Click **OK** to select all of the checked objects.

Selecting Primitive Parent Modules

The Select Primitives Parent command enables you to select the parent modules for all selected primitive logic. To access the command, select **Select Primitive Parents** from the popup menu (available in most views).

Floorplans are much easier to maintain when logic modules are assigned to Pblocks rather than Primitive logic instances. You can easily select a group of timing paths which will in turn select all of the primitive logic instances contained on the paths. The Select Primitive Parents command will automatically select the parent modules for all primitive logic selected.

The originally selected primitive logic is no longer selected. The command interpolates what is selected and returns with only modules selected, unless ROOT level logic was originally selected.

The command will not select parent modules if modules are selected. The pre-selected modules will remain selected.

Using the Select Main Menu Commands

Several of the Selection, Unselect, Highlight and Mark commands are available from the Select menu. For a description of all Select commands, see [“Select Menu” in Appendix A](#).

Fitting the Display to Show Selected Objects

Views in the Workspace have a Zoom option to fit all selected objects. To zoom fit the selected objects, use one of the following methods:

- Select **View > Fit Selection**.
- Press **F9**.
- Click the **Fit Selection** toolbar button.



Figure 4-17: **Fit Selection Toolbar Button**

Using the Selection View

The Selection view, as shown below, displays the list of objects currently selected. Objects may be sorted, unselected or marked from this view. The Selection list is dynamically updated as Floorplan objects are manipulated. To invoke the Selection View, select **Window > Selection**.

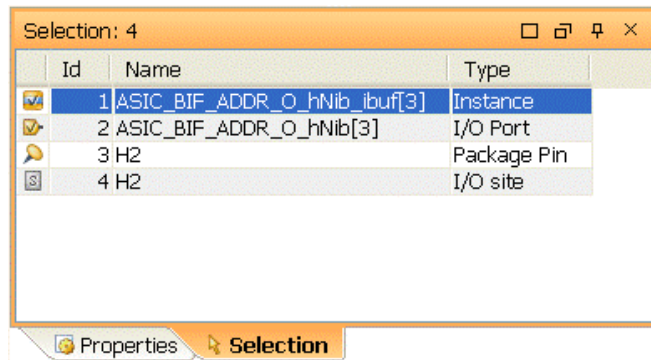


Figure 4-18: Selection View

To sort elements, click on the column header to use as alpha-numeric sort criteria. Objects can be sorted by Name, ID number or Type by clicking on the banner of the desired sort column. Selected items can be removed from the Selection list by using the **Unselect**, **Unselect All** or **Unselect All Except** command from the popup menu.

Groups of objects may be selected by using the **Ctrl** and **Shift** keys. The total number of objects selected is displayed in the view banner.

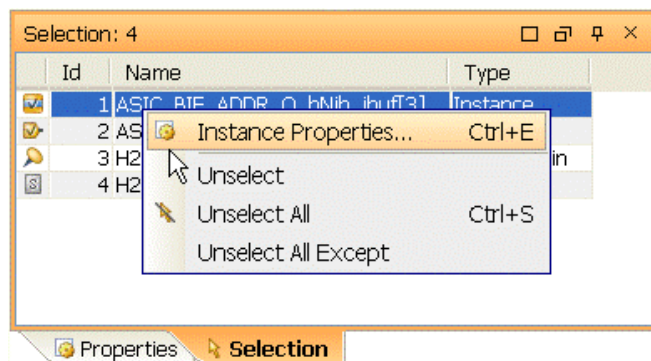


Figure 4-19: Selection View Popup Menu Commands

Highlighting Selected Objects

You can highlight objects with color for display purposes. Highlighting remains until you clear all highlights for the floorplan. For more information on highlighting, see:

- [“Highlighting Selected Objects”](#)
- [“Using the Select Primitives and Highlight Primitives Commands”](#)

Marking Selected Objects

You can place a Mark symbol for all selected objects. For more information, see [“Marking Selected Objects.”](#)

Setting Selection Rules

When selecting an object, other objects may also become selected (e.g. selecting a Pblock also selects the assigned netlist instances). Selection behavior is controlled with selection rules set by selecting **Tools > Options > Selection Rules**.

The Selection Rules dialog box displays.

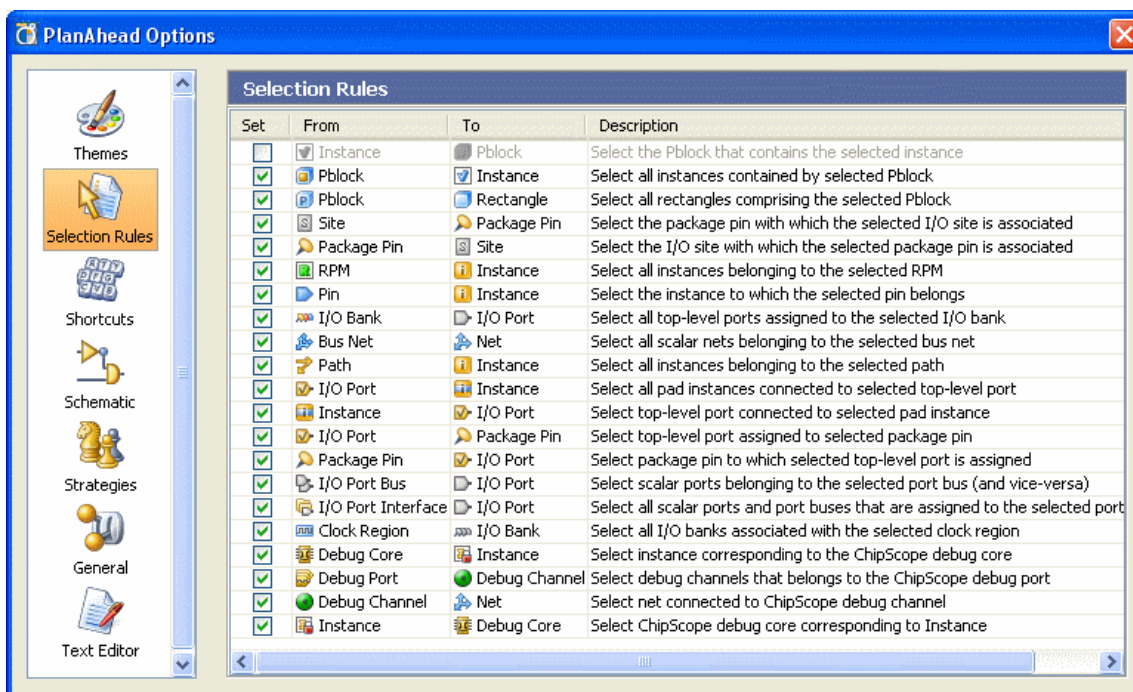


Figure 4-20: PlanAhead Options: Selection Rules

Automatic selection is enabled/disabled by clicking on the Set column heading. Enabling a selection rule will force PlanAhead to select the other affiliated “To” object types when the “From” object gets selected. Disabling the selection rule will force PlanAhead to only select the “From” object when it gets selected.

The default Selection Rules will enable PlanAhead to operate in the most efficient manner. We would advise against changing them unless you know what you are doing.

Setting Selection Ability for Objects in the Workspace Views

Object selection is set in the Themes panel of the PlanAhead Options after selecting **Tools > Options**.

For more information about setting object selections, see [“Customizing PlanAhead Display Options.”](#)

Understanding the Context Sensitive Cursor

The cursor will change the symbol displayed during the command modes described below:

- Pblock edges and Windows view borders may be stretched when the cursor changes into a horizontal, vertical or diagonal stretch bar symbol.
- Pblocks or instances may be moved when the cursor changes into a hand symbol.
- The cursor changes into a cross symbol when rectangles are expected to be drawn for zooming in, defining pin assignment areas or drawing Pblock rectangles.
- When objects are dragged over illegal placement sites, the cursor displays a slashed circle symbol.

- When objects are dragged over legal placement sites, the cursor displays a move point to point symbol.

Configuring the Viewing Environment

PlanAhead has many user configurable viewing options. The tool ships with default settings which can be customized. For more information, see [“Customizing PlanAhead Display Options.”](#)

View layout configurations can be saved and restored for use in subsequent PlanAhead sessions. A separate layout is stored for the PinAhead, Project and Floorplan environments. A layout file is also created to restore the overall PlanAhead window size and location. The view configurations are stored in your home directory when exiting PlanAhead. For more information on the layout of configuration files, see [“Outputs for Environment Defaults.”](#)

Several alternate Floorplan environment layouts are supplied with the PlanAhead software. The available layouts present the most pertinent information in the most productive way. You may save view configurations “Themes” for use in future PlanAhead sessions. For more information, see [“Saving Custom Display Settings.”](#)

Customizing PlanAhead Display Options

View display options can be adjusted to control the appearance and behavior of the environment.

To view or edit the display options, select **Tools > Options**. The PlanAhead Options dialog box will appear. All changes take effect after you select the **OK** or **Apply** button. Clicking **Cancel** will not initiate any changes.

You can view and adjust the settings that control the general environment appearance in the Themes Options. The options can be viewed or changed in the Themes environment for different areas in the interface. The tabs at the bottom of the Themes Options page allow various view setting types to be modified: General, Device, IOs, and Bundle Nets.

Setting General View Display Options

The General tab enables you to customize general color and appearance options of the PlanAhead views.

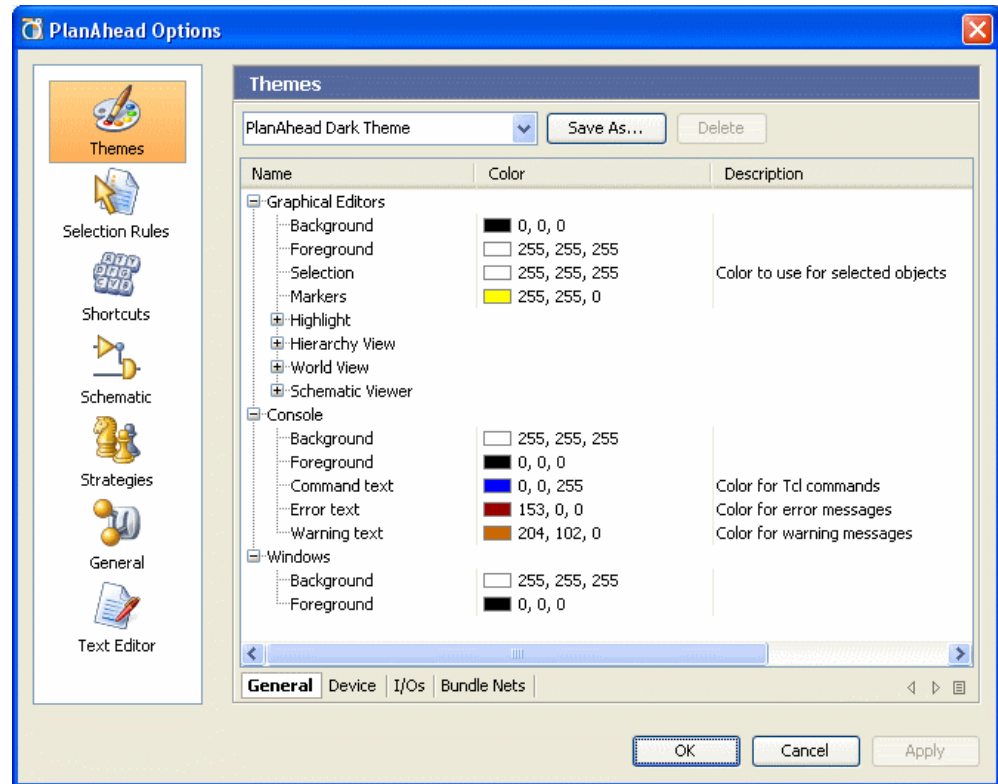


Figure 4-21: General Display Options

Select a color to reveal a pulldown, and click the down arrow to open a popup menu to select another color.

Setting Device View Display Options

The Device tab allows you to adjust the default color, visibility and selection options for each object type in the Device view.

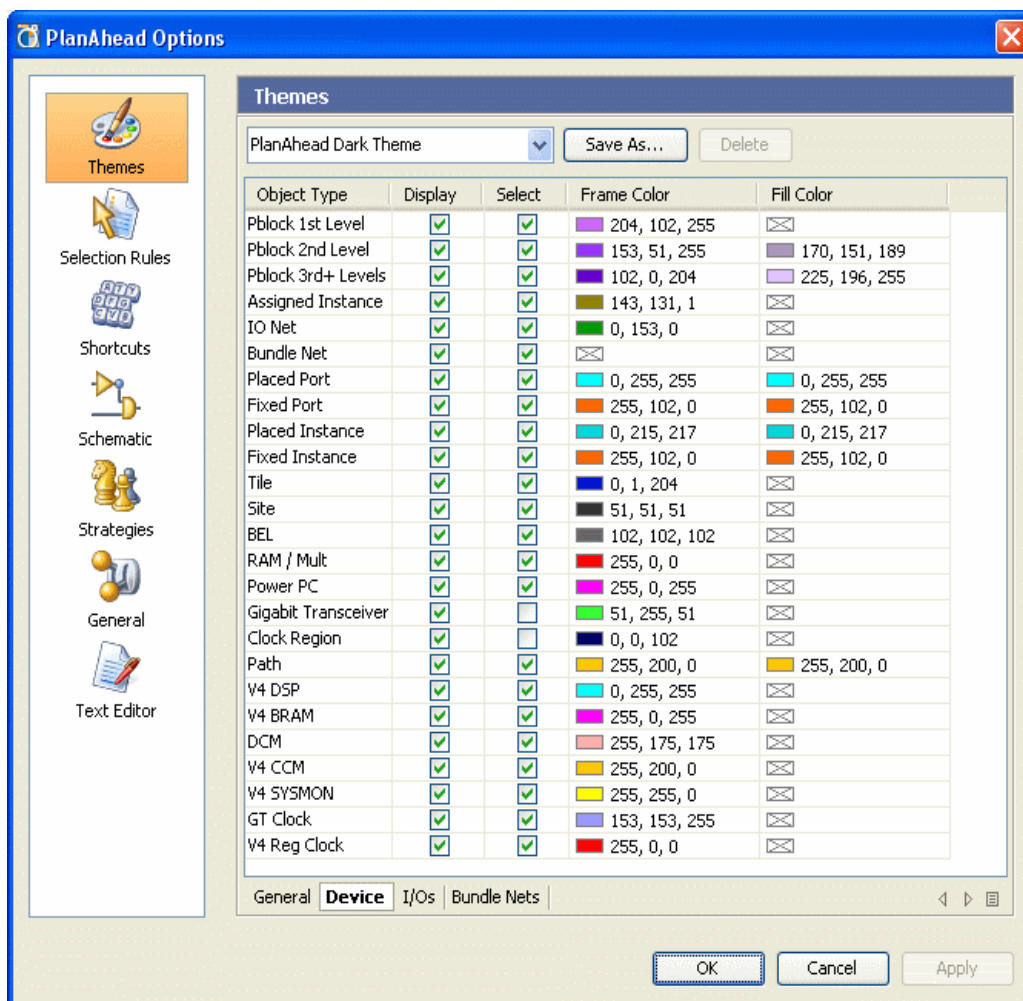


Figure 4-22: Theme Options: Device Settings

The check boxes can be toggled to the desired display effect. In the Display column, toggle the check boxes off to hide the object types in the Device view. In the Select column, toggle the check boxes off to make the object types unselectable in the Device view. They will still be visible if the Display toggle is on.

Note: The Frame and Fill color options are not available for certain object types.

Note: Some of the object types are device specific, so they have no effect in some devices.

Setting Package View Display Options

The I/Os tab allows you to adjust the default color, visibility and selection options for each object type in the Package view.

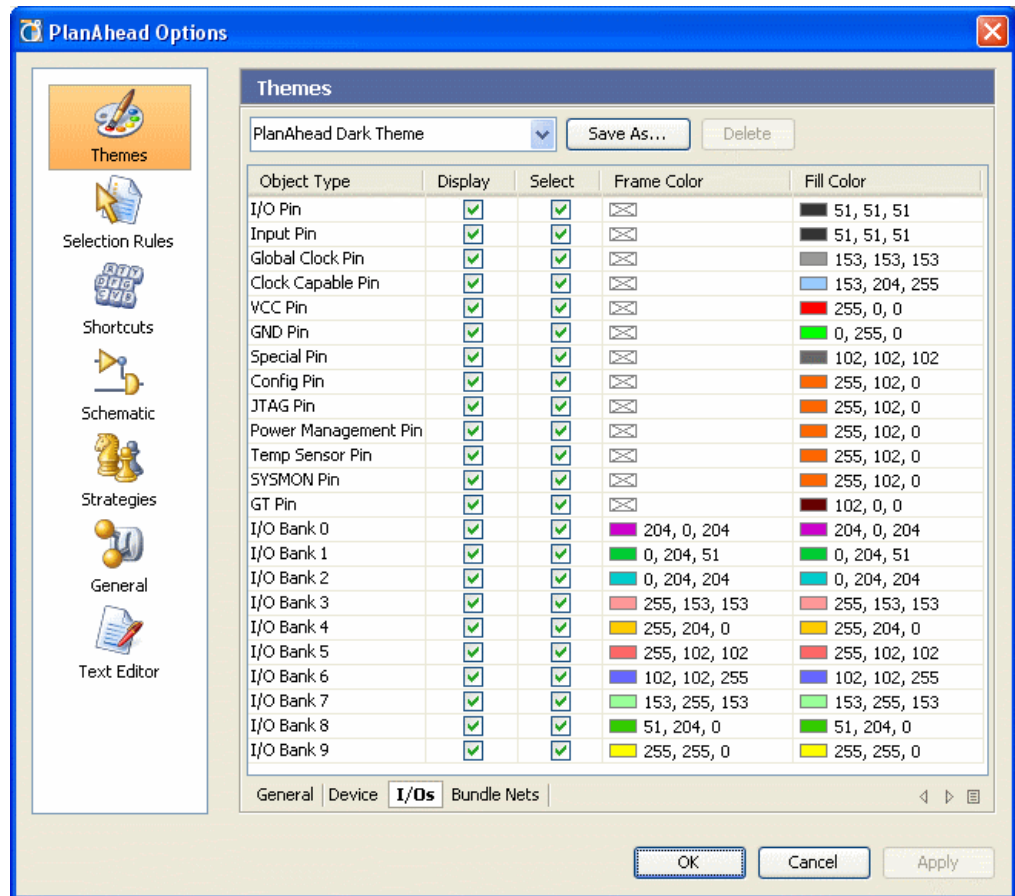


Figure 4-23: Theme Options: Device Settings

The check boxes can be toggled to the desired display effect. In the Display column, toggle the check boxes off to display the object types in the Device view. In the Select column, toggle the check boxes off to make the object types unselectable in the Device view. They will still be visible if the Display toggle is on.

Note: The Frame and Fill color options are not available for certain object types.

Note: Some of the object types are device specific, so they have no effect in some devices.

Setting the Device View Bundle Nets Display Options

The characteristics of the Bundle Nets displayed in the Device view can be configured using the Bundle Nets tab.

The signal count ranges for the desired bundles can be adjusted using the From and To columns. Each column represents a Bundle Net range which can be configured independently.

The check boxes can be toggled for the desired display effect. In the Display column, toggle the check boxes off to hide the Bundle Net range in the Device view. In the Select column, toggle the check boxes off to make the Bundle Nets unselectable in the Device view. They will still be visible if the Display toggle is on.

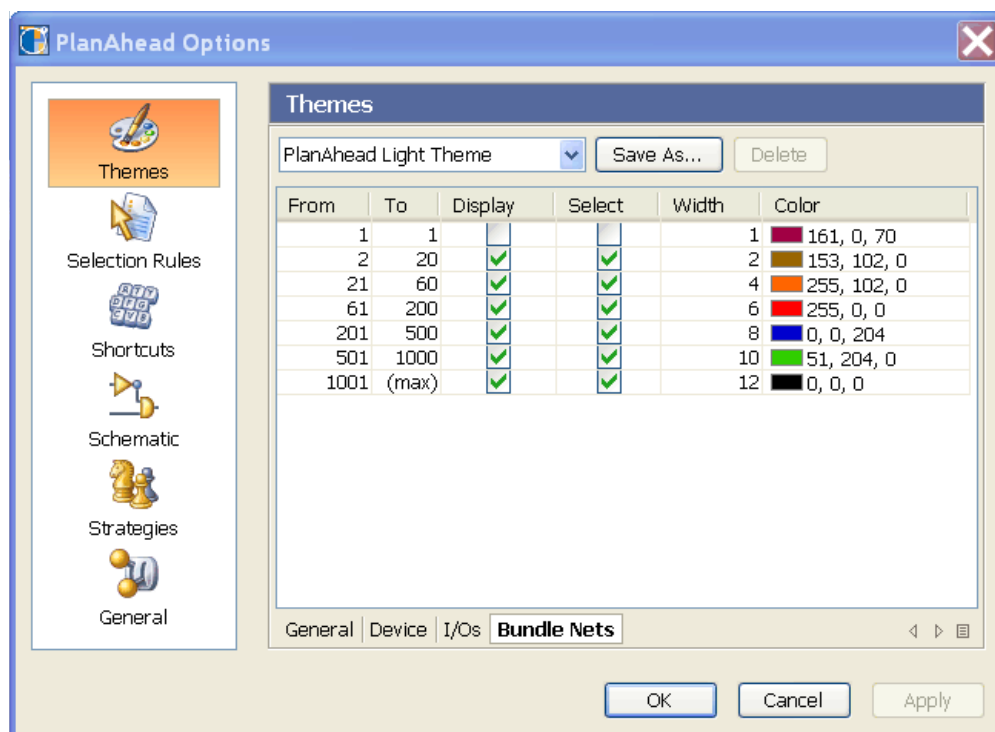


Figure 4-24: Theme Options: Bundle Net Settings

The line width the Bundle appears in the device view can be set for each Bundle Net range by adjusting the values in the Width column.

Configuring Schematic Slack and Fanout Display Options

The Schematic Options enables you to tag source pins with *Fanout* values and destination pins with *Slack* values. For more information, see [“Annotating Slack, Fanout and Values onto Schematic Pins,”](#) page 233.

Adjusting Display using Toolbar Commands

The view display can also be adjusted using the following Device view or Main Toolbar buttons. Some buttons are active only when appropriate object types are displayed. The Toolbar commands are described in [“Select Menu”](#) in Appendix A.

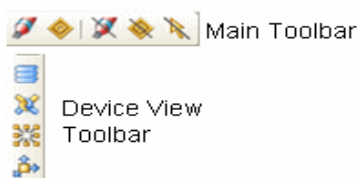


Figure 4-25: View Display Control Toolbar Buttons

You can hover the cursor over the icons to view the tool tip describing the buttons function.

Saving Custom Display Settings

Selecting a Theme

PlanAhead has default view settings for both light and dark background themes. To use either, select the **PlanAhead Light Theme** or **PlanAhead Dark Theme** options in the Theme pull down menu.

These default options are defined in the `planahead.ini` file. For more information, see [“View Display Options File \(planAhead.ini & <theme_names>.patheme\).”](#)

Creating and Using a Customized Theme

You can save your own custom view settings to create PlanAhead initialization files for use in future PlanAhead sessions. To do so, click the **Save As** button next to the Theme pull down menu.

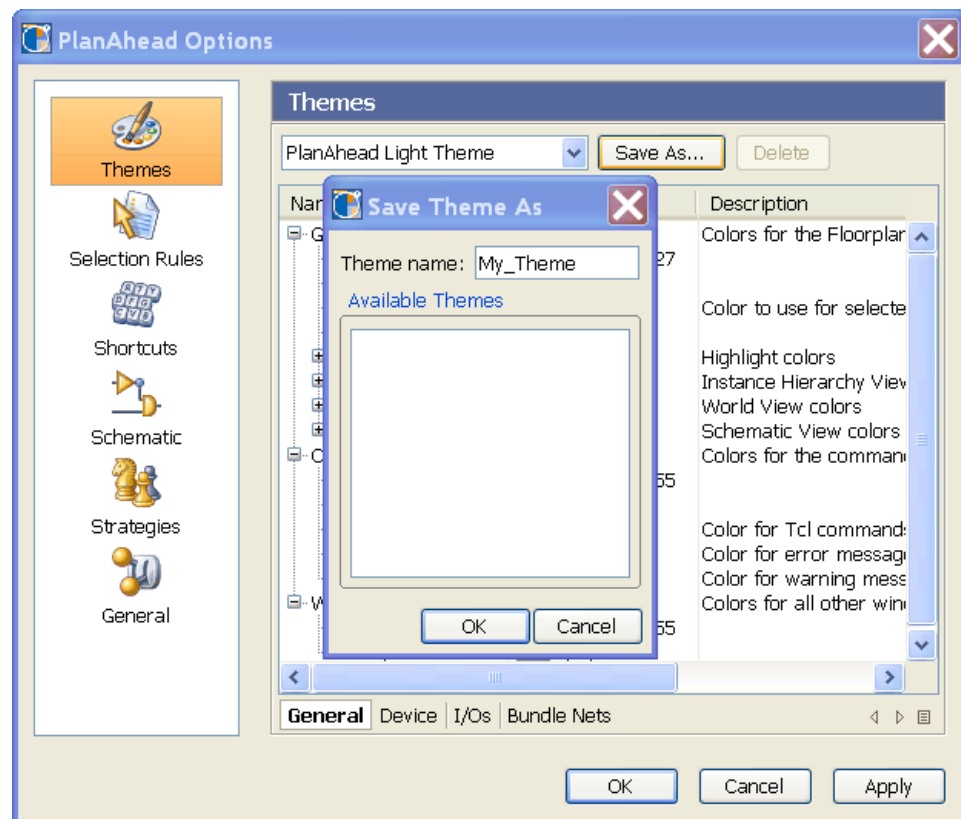


Figure 4-26: Creating a Custom Theme

If you create your own theme, it is a good idea to back up the initialization file that contains the custom settings. For detailed information about the default and custom initialization files, see [“View Display Options File \(planAhead.ini & <theme_names>.patheme\).”](#)

Setting PlanAhead Behavior Options

Setting Selection Rule Options

Selection Rule Options control the object selection settings for all views. When you select an object, other objects may also become selected (e.g. selecting a Pblock also selects the assigned netlist instances). For more information, see [“Selecting Objects.”](#)

Configuring Shortcut Keys

Most commonly used commands also have pre-defined shortcuts using keyboard key combinations. The shortcuts defined are displayed next to the command in the popup menu. For example, the Fit Selection command can be accessed by pressing **F9**.

Default values for shortcuts can be modified by using the Shortcuts option of the PlanAhead Options dialog box.

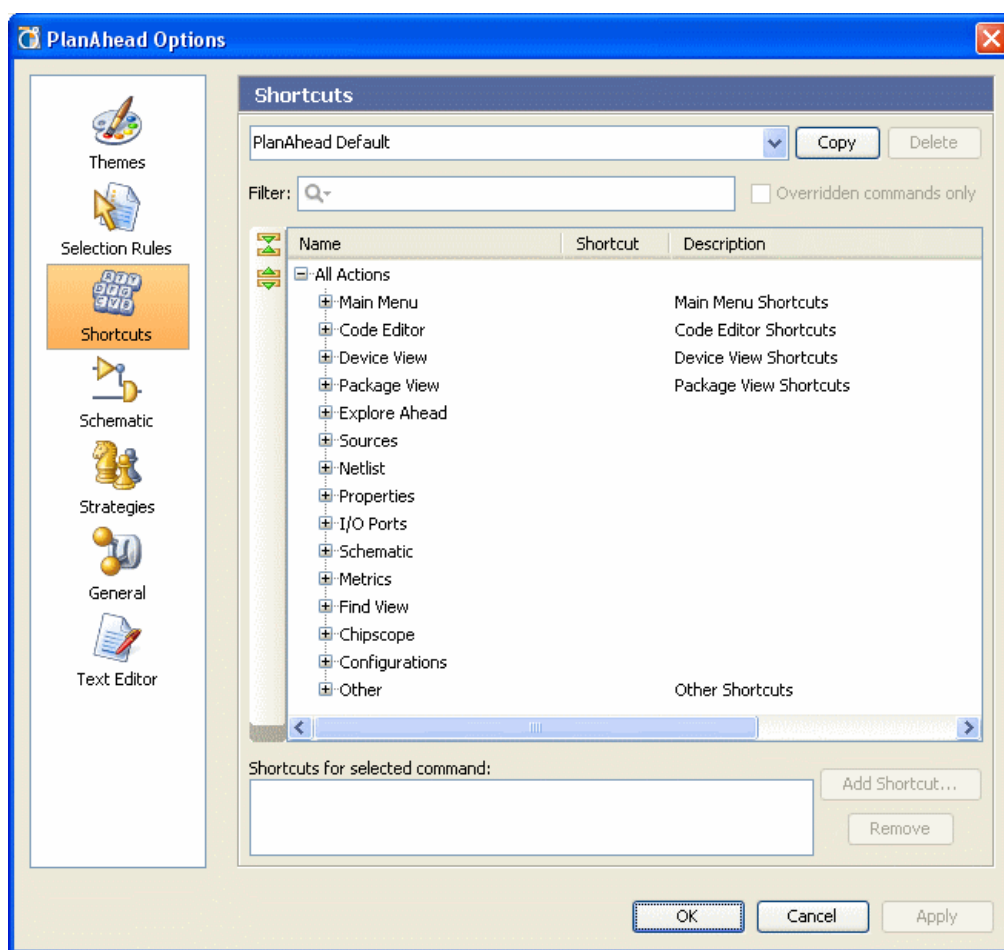


Figure 4-27: Shortcuts Options

The Shortcuts dialog box provides a helpful interface to create new Shortcut schemas which contain custom shortcut settings.

At the top, the Available shortcut schemas enables you to manage Shortcut schemas. Click the **Copy** button to copy the PlanAhead Default schema to create a new schema. You can activate any schema in the list by selecting it from the pull down menu of available

schemas. The PlanAhead Default schema must first be copied in order to make any modifications.

Modifications to shortcuts in the copied schema can be made using the bottom portion of the dialog box. You can search through the list of views and select commands to enter new shortcuts. Select the **Add Shortcut** button and type in the new desired shortcut in the Add Shortcut dialog box. Click **OK** to accept the new shortcut.

Commands listed for shortcut assignment can be filtered using the Filter field containing field. Enter any text string to filter the list of available commands.

Different shortcuts can be used for the same command in different views.

All user-specific shortcut schemas are saved to:

- C:\Documents and Settings\<Username>\Application Data\HDI\shortcuts (Windows)
- ~/.HDI/shortcuts (Linux)

To delete shortcuts, click the **Remove** button.

Setting General PlanAhead Options

Selecting the **Tools > Options > General** tab will invoke the PlanAhead Options dialog box.

Note: The Office 2003 Look and Feel options are not available to Linux users.

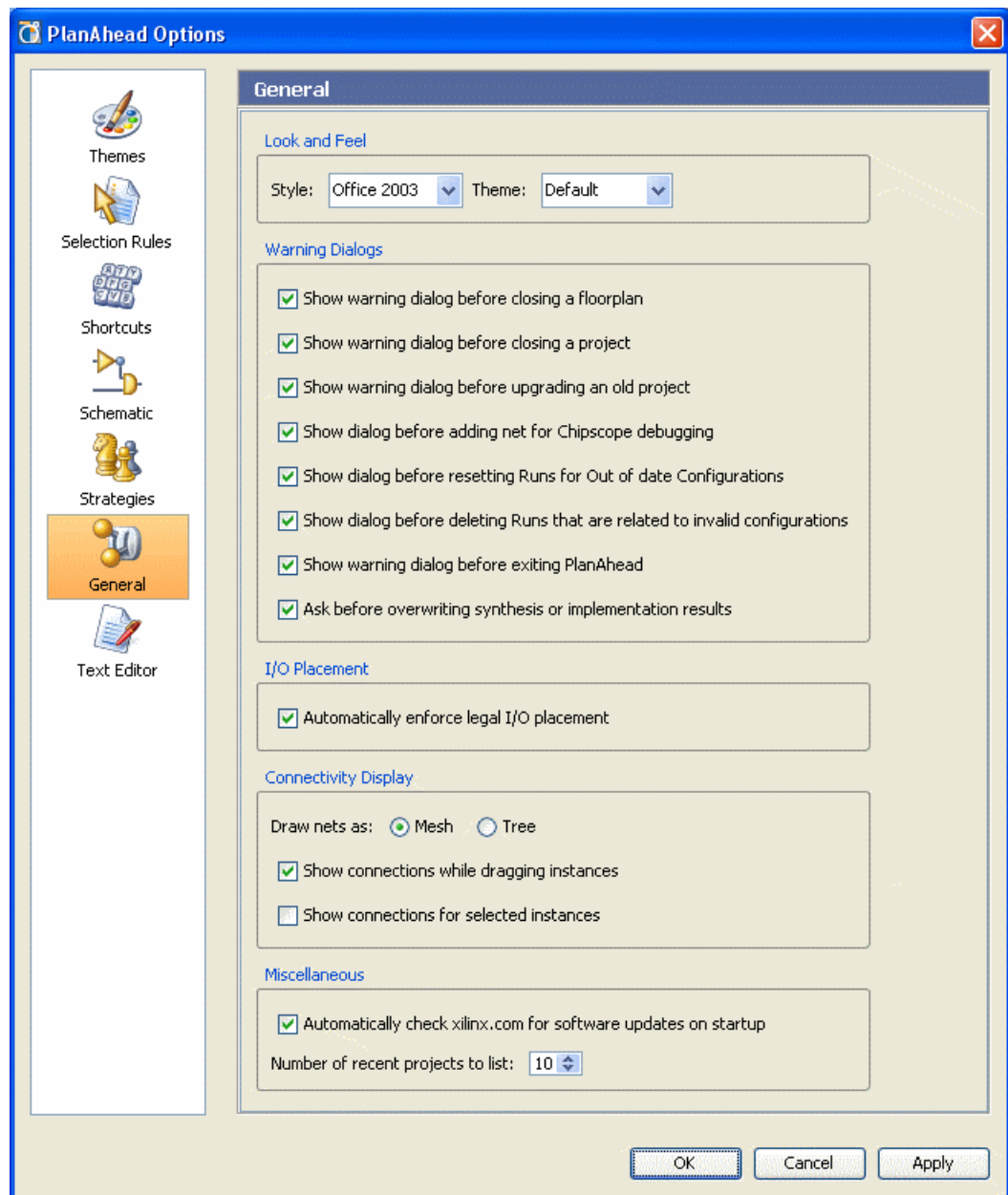


Figure 4-28: General Options

The General options are as follows:

- Look and Feel - You can adjust general style and color options.

Note: Most Windows platform PlanAhead testing and development is done using the default Office 2003 settings. If problems arise with the display, reset the Look and Feel Style back to Office 2003.

- Warning Dialogs - The manner in which warning dialog boxes are presented can be controlled using the self explanatory options.
- I/O Placement - The interactive I/O placement DRCs can be toggled on or off using this option.

- Connectivity Display - The manner in which the connectivity is displayed in the Device view can be controlled using the self explanatory options.
- Miscellaneous - Controls whether PlanAhead automatically searches the Xilinx® website for new software updates, and defines how many previously opened Projects are displayed in the Getting Started view.

Moving Views within Viewing Areas

Multiple views can share the space within the viewing area by displaying them together either vertically or horizontally. Viewing areas can be split by clicking on a view tab and dragging it into another viewing area. An outline will guide you to place the view at the correct position. The resulting window location can be determined prior to accepting it, by watching the moving window outline during the dragging process.

To move a view to share the space in a viewing area:

1. Click the tab, for example the Constraints tab in [Figure 4-29](#).
2. Drag the tab to the desired location. The gray outline will guide you.
3. Release the tab at the desired location.

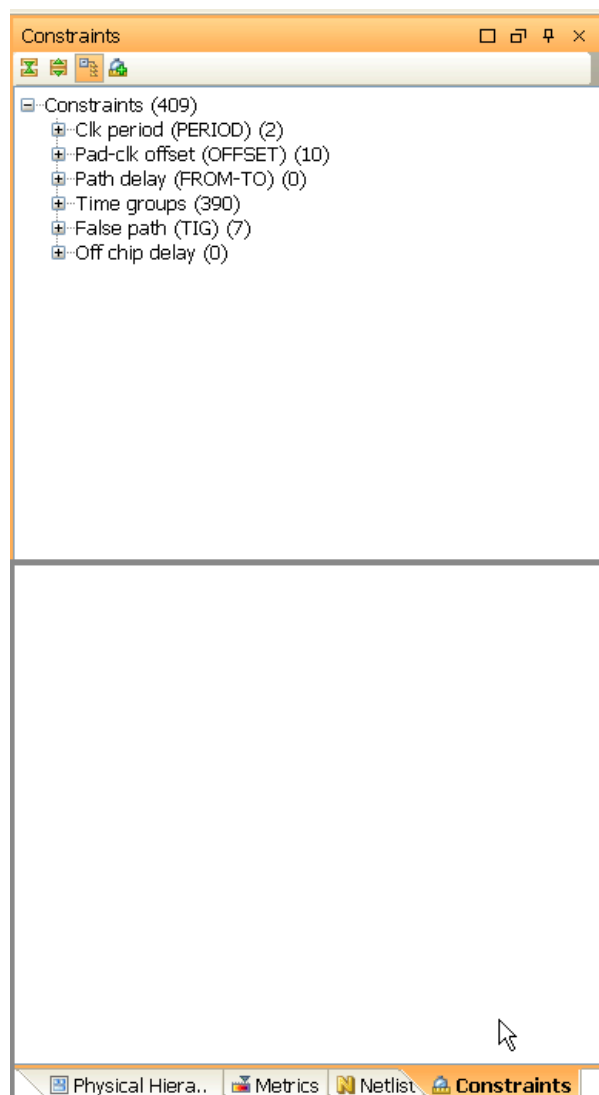


Figure 4-29: Dragging Constraints View to Share View with Other Views

To restore the view to its original location, select **Window > Undo Dragging**. Or, you can repeat the steps above to set the view back in place.

To move a view to a complete different docking area, such as moving the Constraints view to the Properties view, drag the tab of the view you are moving to the banner of the destination docking area.

1. Click the tab.
2. Drag it to the banner of the destination docking area.
3. Release the tab to place the view and its tab.

Creating Custom View Layouts

Select the **Layout** menu commands to save and restore default and alternate view configurations. For a list of Layout menu commands and a description of each, see [“Layout Menu” in Appendix A](#).

Restoring a View Layout

A number of commands are available to restore the layout of PlanAhead.

Views can be docked back to their original locations by toggling the various options previously selected to off again. The view banner manipulation commands also act as toggles and can be selected to revert back to the previous setting.

Restoring the Default View Layout

Select **Layout > Load Layout > PlanAhead Default** to undo any changes to restore the default PlanAhead layout.

Using Undo/Redo Commands

The previous view manipulation commands can be undone by selecting **Layout > Undo**. **Layout > Redo** can be used to repeat a command.

I/O Pin Planning

This chapter contains the following sections:

- “Recommended Method for Pin Planning With Xilinx FPGAs”
- “Using the PinAhead Environment”
- “Viewing Device Resources”
- “Defining Alternate Compatible Parts”
- “Importing I/O Ports”
- “Defining and Configuring I/O Ports”
- “Placing I/O Ports”
- “Placing I/O Related Clock Logic”
- “Removing I/O Placement Constraints”
- “Configuring DCI_CASCADE Constraints”
- “Running I/O Port and Clock Logic Related DRCs”
- “Running Simultaneous Switching Noise (SSN) Analysis”
- “Running Weighted Average Simultaneous Switching Output (WASSO) Analysis”
- “Exporting Package Pin Information”
- “Exporting an I/O Port List”

Recommended Method for Pin Planning With Xilinx FPGAs

The recommended pin planning methodology with Xilinx® FPGA devices is as follows:

1. Address the device configuration mode you plan to use.
Most configuration mode shares some pins with the user I/O. The number of shared pins ranges from a few for serial modes in high pin-count packages to a larger number of for parallel modes and smaller packages. You must avoid signal contention to ensure successful configuration. For more information on device-specific configurations refer to the configuration user guide for the device family with which you are designing. These configuration guides contain detailed information on dedicated and shared configuration pins for each mode.
2. Plan the Gigabit Transceivers.
Gigabit transceivers have a set of dedicated pins and might share clock pins with other Gigabit transceivers or I/O Clock regions. Some device families have a list of user I/Os adjacent to the Gigabit transceivers that you must avoid for optimal signal integrity. More information can be found in Gigabit transceiver user guides.

3. Determine pinout requirement for your memory interfaces.
High speed memory interfaces have specific pinout requirements driven by the memory style width and speed requirements. Xilinx Memory Interface Generator (MIG) generates the required pinouts. Refer to the *Memory Interface Solution User Guide* for more information.
4. Evaluate other IP into your design for pinout requirements.
Some IP, for example the PCI IP, have specific pinout requirements. Use the Xilinx® Core Generator™ tool in the ChipScope™ Pro Core Generator™ software to incorporate designs with the required pinouts. The Core Generator, like MIG, generates the required pinouts for relevant IP.
5. Define additional I/O interfaces.
6. Define required I/O standards and other attributes.
 - a. Define the required I/O standards and any additional I/O attributes for each I/O interface, address the drive strengths, and any other I/O attributes.
 - b. Run DRCs in PlanAhead to check I/O standards against I/O Banking restrictions.
Some I/O standards can be combined within a single Bank; some cannot. Refer to the device-specific packaging and pinout specifications for rules.
7. Define clock pins and clock topology.
 - a. Use the external clock pins on the FPGA for best clock performance.
 - b. Understand the I/O versus the fabric clocking resources, and any device-specific restrictions on regional clocks.
Generally, pinout restrictions are few for simpler clock structures and lower clock counts (fewer than the number of global and I/O clocks on the device). For more complex clock structures; for example, high clock counts that require either automatic or manual floorplanning to use regional clocks, you must enter the clock tree and enough loads to be able to run the early design through ISE® for validation. For more information regarding clock pins and clock topology refer to the device-specific Clocking user guides.
8. Run the design through ISE to validate final pinouts.
The more of the design I/O, IP, and clocking structures that are available, the more accurate DRCs are for the design. You do not need to have all the logic in place, just the primary structures (as described in the previous steps) that affect pinout.
9. Analyze pinouts for Simultaneous Switching Noise (SSN) or Weighted Average Simultaneous Switching Output (WASSO), based on device family.
 - SSN is available for the Virtex®-6 FPGA device in the PlanAhead™ software. See [“Running Simultaneous Switching Noise \(SSN\) Analysis,” page 157](#) for more information.
 - Weighted Average Simultaneous Switching Output (WASSO) is supported in Virtex-5 and Spartan®-3 device families through rules published in the device-specific select I/O User Guides. See [“Running Weighted Average Simultaneous Switching Output \(WASSO\) Analysis,” page 161](#) for more information.
10. Perform signal integrity analysis using IBIS or HSPICE models for production boards.

PinAhead Overview

The PinAhead environment provides an interface to analyze the design and device I/O requirements, and to define an I/O pinout configuration or “pinout” that satisfies the requirements of both the PCB and the FPGA designers. The PlanAhead software enables the creation of I/O port signals, and the import of an I/O port list in CSV, UCF, or HDL format. This allows for early and intelligent pinout definition to eliminate pinout-related changes that typically happen later in the design. It can also substantially improve the performance. Often, designers are hindered by a non-optimal pinout that causes further delays when trying to meet timing and signal integrity requirements. By considering the data flow from PCB to FPGA die, optimal pinout configurations can be achieved quickly, thus reducing internal and external trace lengths as well as routing congestion.

Using the PinAhead Environment

Many of the views available in the PinAhead environment are also used in the Floorplan environment. For more information about the Floorplan environment, see [“Floorplan Environment View Layout.”](#)

The PinAhead environment consists of a split Workspace showing both the Package and Device views. There are other views that provide additional I/O information: the Clock Region view, Package Pins view, and the I/O Ports view.

There are two ways to launch the PinAhead view layout:

- Select **Tools > Open PinAhead**.
- Create a new empty Project using the New Project wizard as shown in [Figure 5-1, page 128](#).

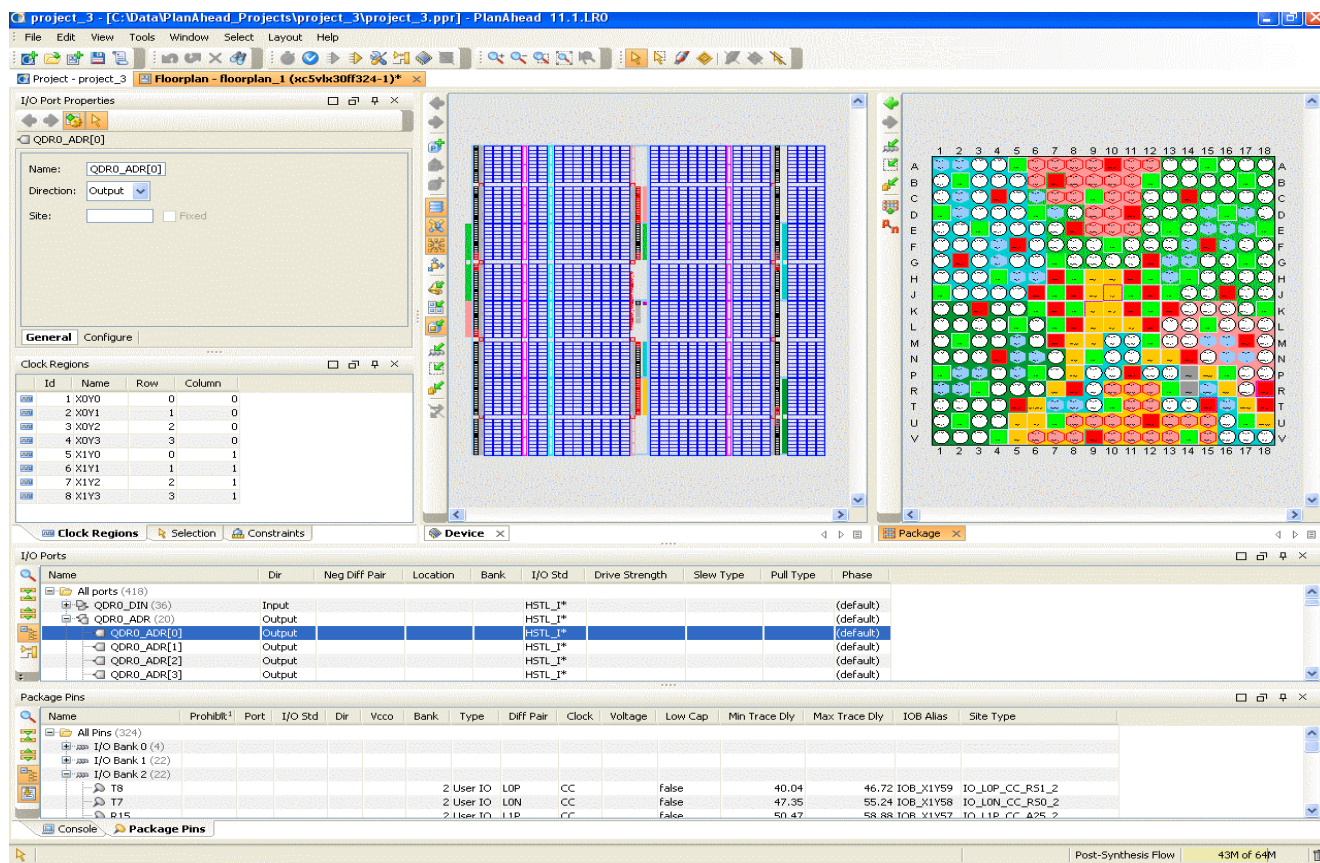


Figure 5-1: PinAhead Environment

The I/O Ports view displays the I/O signals that are currently defined in the design. Busses are grouped and displayed in bus folders. Differential pair signals and busses are also grouped.

You can create Custom Interfaces by selecting groups of signals and/or busses, and then selecting the **Create I/O Interface** command. Additional commands are available for modifying existing interface content also, as described in [“Creating I/O Port Interfaces.”](#)

You can assign the individual pins, busses, or interfaces to I/O pins by dragging them into either the Device or the Package view. The entire group of pins is assigned to I/O pins using various assignment pattern modes. The modes available include Place I/O Ports in an I/O Bank, Place I/O Ports in Area, and Place I/O Ports Sequentially.

Each mode offers a different assignment pattern for the I/O Ports to be assigned to pins. Information about the number of ports being placed is provided on the cursor tool tip. The mode remains active until all of the selected I/O ports are placed or until you press the **Esc** key. For more information, see [“Placing I/O Ports.”](#)

The software attempts to maintain correct assignment rules. Differential pair ports are assigned into proper pin pairs. Online DRCs are also available to help ensure legal I/O placement. The batch PlanAhead I/O related DRCs flag such I/O issues.

The **Tools > Autoplace I/O Ports** command places the entire device or any selected portion of the device automatically. The command obeys I/O bank rules, differential pair rules, and global clock pins, and places as many of the I/O Ports as possible.

The command also attempts to group the interfaces as much as possible. You can place Prohibits on individual I/O pins or I/O banks to prevent I/O assignment to them.

You can select and configure any of the Ports or Interfaces using the Configure I/O Ports command. This command provides a way to set I/O standard, drive strength, and slew type. PinAhead supports output to a comma separated values (CSV) format file for use in PCB schematic symbol creation or the HDL port list.

Using the I/O Ports View

The I/O Ports view shows the I/O signal ports defined in the design. These ports can be defined within PlanAhead by importing an EDIF netlist or by importing a CSV I/O port list. To invoke the I/O Ports view, select **Window > I/O Ports**. The following figure shows the display.

Name	Dir	Neg Diff Pair	Location	Bank	I/O Std	Drive Strength	Slew Type	Pull Type	Phase
All ports (418)									
QDR0_DIN (36)	Input				HSTL_I*				(default)
QDR0_ADR (20)	Output				HSTL_I*				(default)
QDR0_DOUT (36)	Output				HSTL_I*				(default)
QDR1_DIN (36)	Input				HSTL_I*				(default)
QDR1_ADR (20)	Output				HSTL_I*				(default)
QDR1_DOUT (36)	Output				HSTL_I*				(default)
DIRAC_SPARE (4)	In/Out				LVCMS525	12 SLOW			(default)
FRM0_SBFC (3)	Output				LVTTTL*	12 SLOW			(default)
FRM1_SBFC (3)	Output				LVTTTL*	12 SLOW			(default)
LED (4)	Output				LVCMS525	12 SLOW			(default)
FRM0_RDat_P (16)	Input				LVDS_25				(default)
FRM0_RStat (2)	Output				LVTTTL*	12 SLOW			(default)
FRM0_TDat_P (16)	Output				LVDS_25				(default)
FRM0_TStat (2)	Input				LVTTTL*	12 SLOW			(default)
FRM1_RDat_P (16)	Input				LVDS_25				(default)
FRM1_RStat (2)	Output				LVTTTL*	12 SLOW			(default)
FRM1_TDat_P (16)	Output				LVDS_25				(default)
FRM1_TStat (2)	Input				LVTTTL*	12 SLOW			(default)
PCI_AD (32)	In/Out				PCI33_3*				(default)
PCI_CBE (4)	Input				PCI33_3*				(default)
PCI_CBE[0]	Input				PCI33_3*				(default)
PCI_CBE[1]	Input				PCI33_3*				(default)
PCI_CBE[2]	Input				PCI33_3*				(default)
PCI_CBE[3]	Input				PCI33_3*				(default)
Scalar ports (48)									
CORE_CLK_P	Input	CORE_CLK_N			LVDS_25				(default)
CORE_RST	Input				LVCMS525	12 SLOW			(default)
DELAYREF_CLK_P	Input				LVTTTL*	12 SLOW			(default)
QDR0_KP	Output				HSTL_I*				(default)
QDR0_KN	Output				LVCMS525	12 SLOW			(default)

Figure 5-2: I/O Ports View Displaying I/O Port Interfaces

The view displays the ports according to category by Interface, or displays the ports alphabetically by clicking the **Group by Interface or Bus** toolbar button in the I/O Ports view.



Figure 5-3: Group by Interface or Bus Toolbar Button

Port signal names, direction, package pin, bank, I/O Standard, Drive strength, Diff pair partner, Slew type and other signal information are listed for each I/O port. Table values

appear black if they are default values, black with an asterisk for non-default values, and red when they are illegal or undefined values.

Ports can be selected and grouped together into Interfaces by selecting the **Create I/O Port Interface** popup menu command. These Interfaces can be selected and placed as one object within the PinAhead environment.

Busses are displayed as folders in the view. Each folder is a group of bus signals.

Ports and Interfaces can be selected from the I/O Ports view and assigned using the PinAhead environment.

Using the Package Pins View

The Package Pins view displays device package pin information. Information about the device pins is displayed in table form. To invoke the Package Pins view, select **Window > Package Pins**.

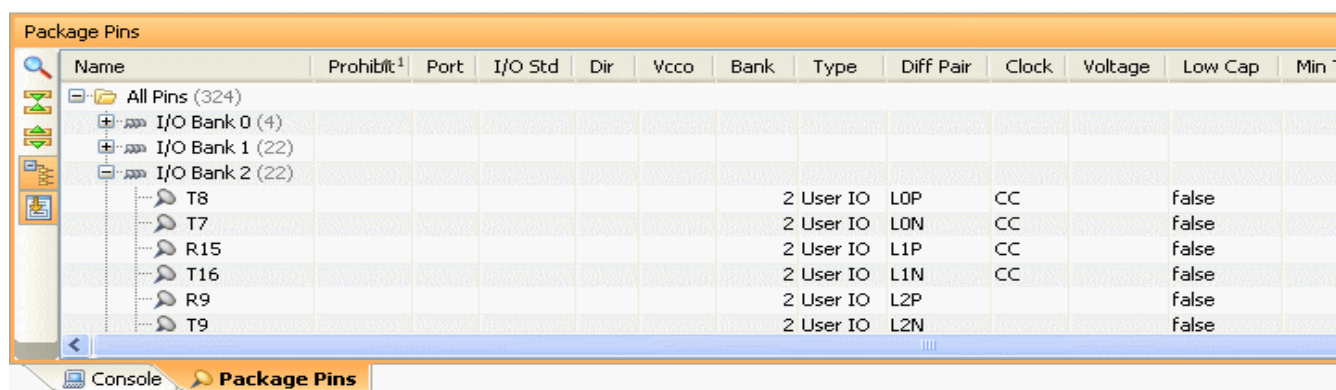


Figure 5-4: Package Pins View

The view can be toggled to display pins according to category by I/O bank, or displays the pins alphabetically by clicking the **Group by I/O Bank** toolbar button in the Package Pins view.



Figure 5-5: Group by I/O Bank Toolbar Button

Device pin information such as I/O Bank number, Type, Differential pair partners, Site Types and Min/Max package delay are listed for each package pin. Table values appear gray when they are the default values, black when they are non-default values, and red when they are illegal values.

Note: The unit of measurement for the Min/Max package trace delay in the Package Pins view is in picoseconds (ps).

The information in the Package Pins view can be sorted by clicking any of the column headers. Clicking again will reverse the sort order. You can sort by a second column by pressing the **Ctrl** key and clicking another column to perform a secondary sort. Add as many sort criteria as necessary to refine the list order.

Using the Package View

The Package view displays the physical characteristics of the device being used in the design. Various types of pins are displayed using different colors and shapes

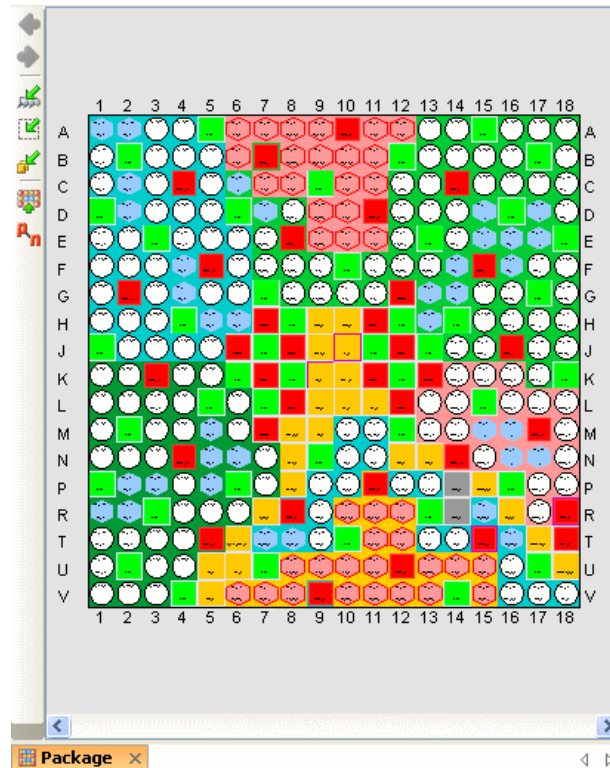


Figure 5-6: Package View

You can open the Package view by selecting the Package view tab in the Workspace or by selecting **Window > New Package View** command. Multiple Package views can be opened simultaneously.

Moving the cursor within the Package view actively displays the I/O pin coordinates on the top and left sides of the view. Additional I/O pin and bank information is displayed in the Status bar located at the bottom of the PlanAhead Environment. The active object being reported is highlighted in the Package view.

Holding the cursor over the Package view invokes a tool tip that displays the pin information.

Ports and I/O buffer instances can be dragged into the Package view for assignment. They can also be reassigned to other I/O pins within the Package view.

VCC and GND pins are displayed as red and green square pins. Clock capable pins are displayed as hexagon pins.

The colored areas between the pins display the I/O banks. You can select Pins or banks by clicking them with the mouse. Select I/O pins or banks to highlight them in the Device view. Pins or I/O banks that are selected in the Device view are also highlighted in the Package view.

You can display the differential pair pins in the Package view by toggling on the **Show Differential I/O Pairs** toolbar button.



Figure 5-7: **Show Differential I/O Pairs Toolbar Button**

The Package view can be made to appear from the top or bottom of the package by clicking the **Show Bottom/Top View** toolbar button, or selecting the **Show Bottom/Top View** popup menu command.



Figure 5-8: **Show Bottom/Top View Toolbar Button**

There are several icons in the upper left corner of the Device view. These view-specific Toolbar commands are covered in [“Using Common Popup Menu Commands,” page 251](#).

Printing the Package View

You can print the Package view using the **File > Print** command. The current viewable area is printed. To print the entire Package view, zoom to fit and then print.

Using the Device View

The I/O Pad, Banks and Clock related resources are displayed in the Device view. Users can cross select I/O resources in the any of the PinAhead views to view the relationship between the physical package pins and the internal die resources. For more information, see [“Using the Device View” in Chapter 8](#).

Viewing Device Resources

Viewing Package Pin Properties

Pins or I/O banks can be selected in the Package view and corresponding details are displayed in the Properties view. Select an object in the Package view, and view the details in the Properties view as shown in [Figure 5-9, page 133](#).

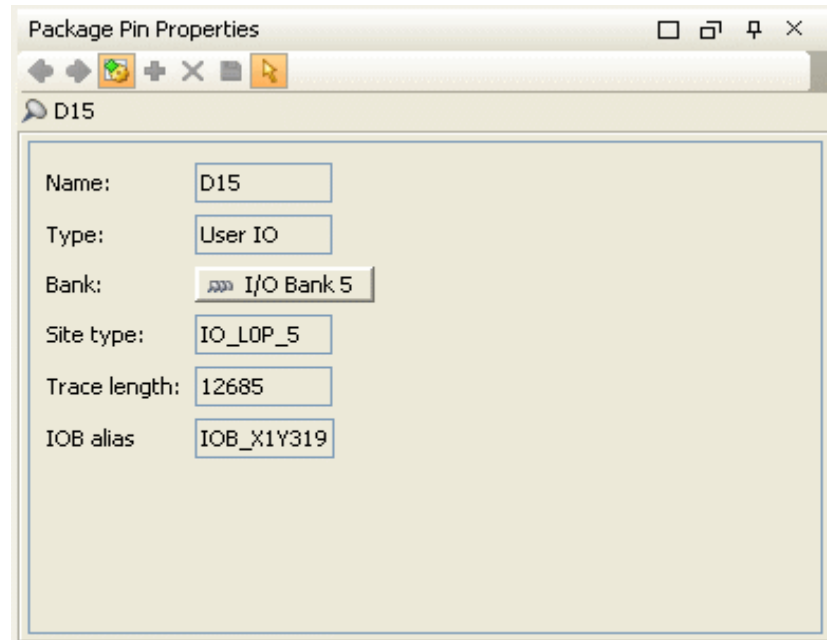


Figure 5-9: Package Pin Properties

Viewing I/O Bank Resources

I/O resources can be selected in any of the PinAhead views and their corresponding data is highlighted in all other views. This provides a visual indication of the relationship between the physical package and the internal die. Many types of device and design information is displayed in the various view. There are extensive exploration possibilities, such as:

- Select one of the I/O banks in the Package Pins view.
- Click the General Tab in the I/O Bank Properties view.

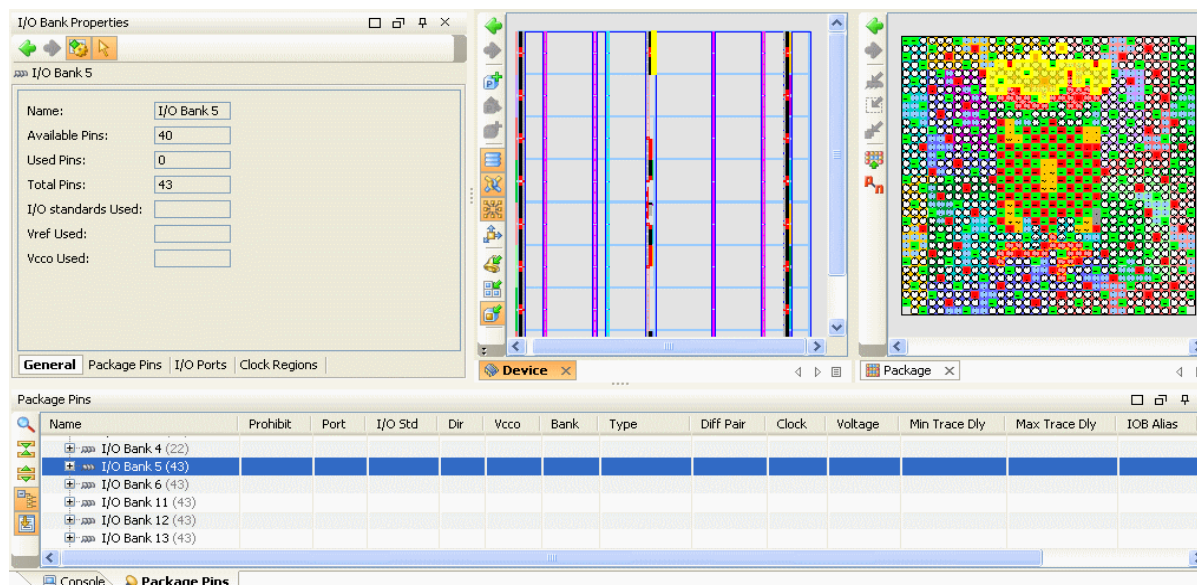


Figure 5-10: Displaying I/O Bank Location and Resources

Viewing Clock Region Resources

The clock regions are displayed graphically in the Device view. They are currently shown as dark blue rectangles, but the color selection can be modified in the Device dialog box (accessible using **Tools > Options > Themes > Device**).

The Clock Regions view allows easy selection of the clock regions. Selecting a clock region will highlight the related I/O banks and regional clock resources. The Clock Region Properties view displays the resources statistics and logic content of the selected clock region.

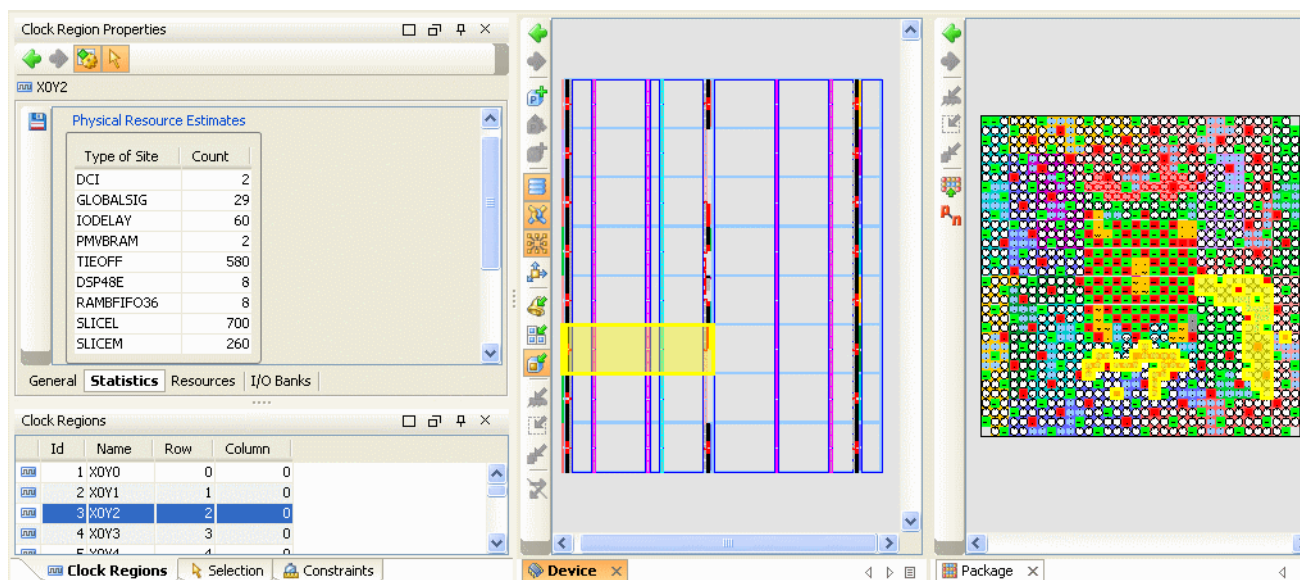


Figure 5-11: Viewing Clock Region Resources

Defining Alternate Compatible Parts

PlanAhead enables you to select an alternate device for the design. PlanAhead will attempt to ensure that a legal I/O pin assignment is defined and will work across all of the selected devices.

1. Select the **Make Part Compatible** popup menu command in the Package view.

A list of available compatible devices available in the same package are displayed. This feature only supports alternate parts available in the same package.

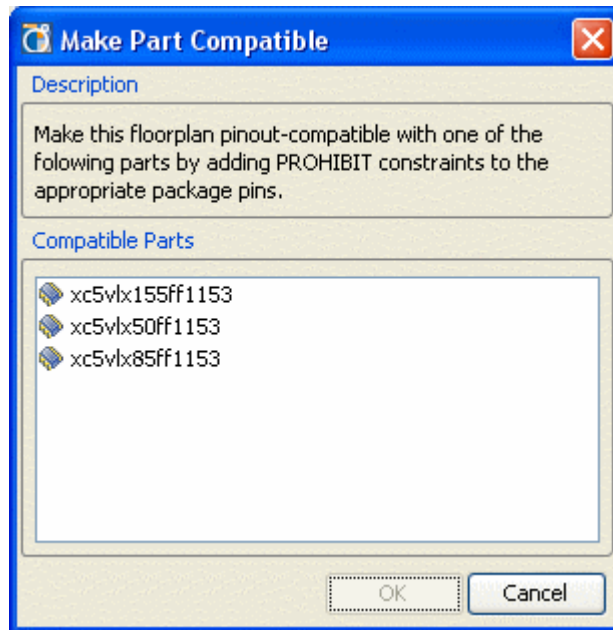


Figure 5-12: Select Alternate Compatible Parts

2. You can select any number of alternate parts realizing that the number of available Package Pins for placement might diminish as more parts are selected.

Prohibits are placed automatically on any unbonded pins in the alternate devices selected. A dialog box displays showing the number of package pins prohibited. Prohibits placement is dependent on the alternate parts selected.

The Prohibits can also be viewed in either the Package, Package Pins or Device views. The Alternate Parts defined can be viewed and managed in the Floorplan Properties view under the Part Compatible tab.

Note: The Make Part Compatible command supports Virtex®-5 and Virtex-6 parts only at this time.

The alternate parts defined for a floorplan can be displayed and modified using the Add Compatible Part toolbar button in the Floorplan Properties view.

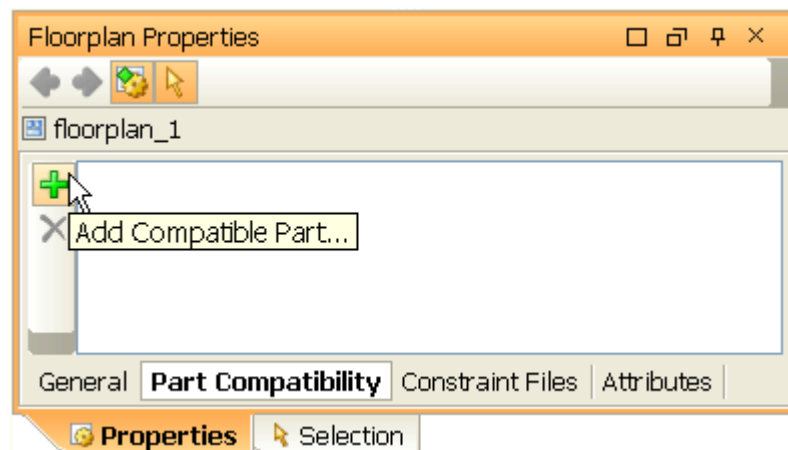


Figure 5-13: Add Compatible Part toolbar button

Importing I/O Ports

PlanAhead supports importing UCF, CSV or HDL header format files into an empty PlanAhead Project to begin I/O pin planning.

Creating an RTL based or synthesized netlist based Project will automatically populate the I/O Ports view with the I/O ports defined in the design.

CSV Format File

To import an I/O ports list from a CSV:

1. Make sure you have a Floorplan in focus.
2. Select **File > Import I/O Ports > From CSV**.
3. In the file browser, browse to and select a CSV file to import.

The CSV file format is shown in [Figure 5-14, page 136](#). At a minimum, the Signal Name field must be present in the file.

	A	B	C	D	E	F	G	H	
1	Top: top Floorplan: floorplan_1 Part: xc5vsx35tf1665-1								
2	Generated by: brianj on: Fri Feb 06 17:28:39 2009								
3	Build: PlanAhead v11.1.LR0 by: ECloudInternalUser4 on: Thu Feb 5 20:04:57 PST 2009								
4									
5	IO Bank	Pin Number	IOB Alias	Site Type	Min Trace	Max Trace	Prohibit	Interface	Sig
6		P2	OPAD_XD05	MGTTXP0_114	34878	40691			TXP
7		W2	OPAD_XD07	MGTTXP1_114	41406	48307			TXP
8		B2	OPAD_XD013	MGTTXP0_116	63540	74130			TXP
9		G2	OPAD_XD015	MGTTXP1_116	55620	64890			TXP
10									

Figure 5-14: I/O Port List CSV Format

CSV Columns

CSV is a standard file format used by FPGA and board designers to exchange information about device pins and pinout.

- **I/O Bank** – The I/O Bank in which the pin is located. The software fills in this field for all pins in the device. Values are a number or blank. This is not required in the input CSV file.
- **Pin Number** – The name (or location) of the package pin. The software writes this out for all pins in the device. This is not required in the input file. If used for input, it is used to define placement. Values are legal pins in the device.
- **IOB Alias** – The alternate part name for the package pin. This field is specified by the software, and is unused if specified in the input CSV file.
- **Site Type** – The name for the pin from the device data sheet. This field is specified by the software, and is unused if specified in the input CSV file.
- **Min/Max Trace Delay** – The distance between the pad site of the die and the ball on the package, in picoseconds. This is specified by the tool to help the board engineer match trace delays. The Trace Delay fields are in the output file only. They are not expected in the input file.
- **Prohibit** – Certain sites may be prohibited for many reasons to prevent user I/O from being added to the site.

Prohibits ease board layout issues, reduce cross-talk between signals, and ensure that a pinout will work between multiple FPGAs in the same package. In the UCF this is represented by a `CONFIG PROHIBIT` constraint. Values are `TRUE` or the field is left blank. This field should be left blank when the Pin Number is left blank.

- **Interface** – A user-specified grouping for an arbitrary set of user I/O. As an example, this field provides a means to specify a relationship for the data, address, and enable signals for a memory interface. Values are a text string or blank. This field is optional.
- **Signal Name** – The name of the User I/O in the FPGA design. Values are a string or blank for an unassigned Package Pin.
- **Direction** – The direction of the signal. Values are `IN`, `OUT`, `INOUT`, or blank when a user I/O is not assigned to the site.
- **DiffPair Type** – Instructs the software about which pin is the N side of a differential pair, and which pin is the P side. This is only used for differential signals. The software uses this column instead of a naming convention to figure out which pin is the N side of the pair, and which pin is the P side. Values are `P`, `N` or blank when a user I/O is not assigned to the site.
- **DiffPair Signal** – Used to specify the name of the other pin in the differential pair. Values are the name of the user I/O or blank when unused.
- **I/O Standard** – The I/O standard for a specific user I/O. When this field is blank for a user I/O, the software uses the appropriate device defaults. Values are a legal I/O standard for the user I/O in the device or blank.
- **Drive** – The drive strength of the I/O standard for a specific user I/O. Not all I/O standards accept a drive strength. If this field is blank the tools will use the default. Values are a number or blank.
- **Slew Rate** – The slew rate of the I/O standard for a specific user I/O. Not all I/O standards accept a slew rate. If this field is blank the tools will use the default. Values are `FAST` and `SLOW`.
- **Phase** – Used to specify the phase of an I/O relative to the phase of other I/O in the bank in cases of a synchronous phase offset.

You can also attach other information. The software will add any other fields with user-defined values to the set of user-defined columns. Also, you can add extra columns to the Package Pins table.

Using User Defined I/O Port Properties in the CSV

PlanAhead has a specific expected format for importing I/O pin related data. Often, design groups have additional information in their CSV files. If additional columns of data exists in the imported CSV file, PlanAhead will create new columns in the Package Pins view to display and/or modify the field values. Select the **Set User Column Values** popup menu command to modify or define values in the customer CSV fields. If you select **File > Export I/Os > CSV**, the columns and new values are preserved and exported in the output CSV file.

HDL Format Files

PlanAhead lets you import an HDL format header as a way to populate the I/O Ports view. To import I/O port definitions from HDL files, select **File > Import I/O Ports > From HDL**. In the wizard, you can specify VHDL and/or Verilog files for import. Enter the top-level module name and the search paths to the HDL files and libraries. PlanAhead will then import the I/O ports from the HDL files.

UCF Format File

PlanAhead lets you import a UCF format files as a way to populate the I/O Ports view. To import I/O port definitions from a UCF file, select **File > Import I/O Ports > From UCF**.

Because the UCF format does not define port direction, the *Direction* fields will display *<undefined>*. Select the **Set Direction** command from the I/O Ports popup menu to define I/O port direction. For more information, see [“Setting I/O Port Direction.”](#)

Defining and Configuring I/O Ports

I/O ports can be created and configured using the PinAhead interface. Empty Projects can be created, and an I/O port list can be easily generated.

Creating I/O Ports

To create I/O ports:

1. In the I/O Ports view, select **Create I/O Ports** from the popup menu.
The Create I/O Ports dialog box opens.

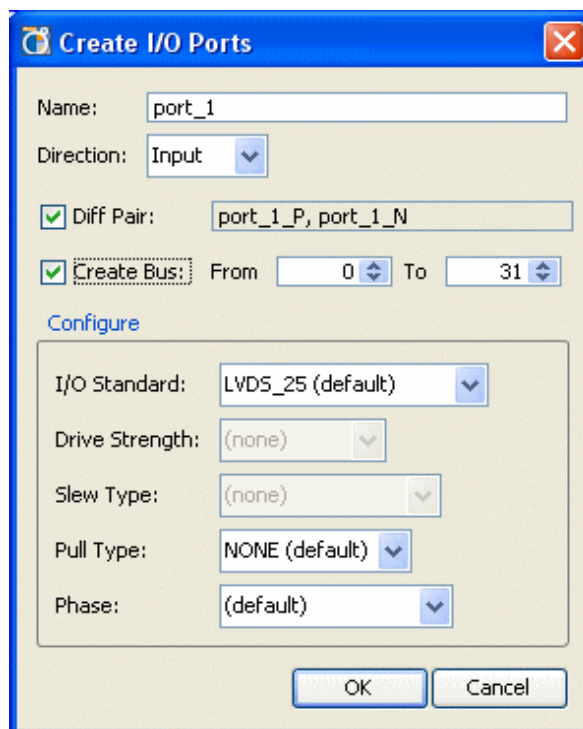


Figure 5-15: Create I/O Ports dialog box

2. View and edit the options in the Create I/O Ports dialog box:
 - **Name**—Enter the name of the desired port or bus to create.
 - **Direction**—Select the desired port direction.
 - **Diff Pair**—Define differential pair signals or busses.
 - **Create Bus**—Enter bus range for bus creation.

- **Configure**
 - **I/O Standard**—Select the desired I/O Standard constraint.
 - **Drive Strength**—Select the desired Drive Strength value.
 - **Slew Type**—Select the desired Slew Type value.
 - **Pull Type**—Select the desired Pull Type value.
 - **Phase**—Enter a phase group or select an existing phase group. A phase group is a logical grouping of ports that is used in SSN calculations to indicate that the set of ports share the same frequency and phase. For more information on using this option, see [“Defining the I/O Port Switching Phase Groups,”](#) page 160.

Refer to Xilinx device documentation for information regarding voltage capabilities of the device.

Configuring I/O Ports

To configure a port or a group of ports:

1. Select the ports in the I/O Ports view.
2. Select **Configure I/O Ports** from the popup menu.

The Configure Ports dialog box opens.

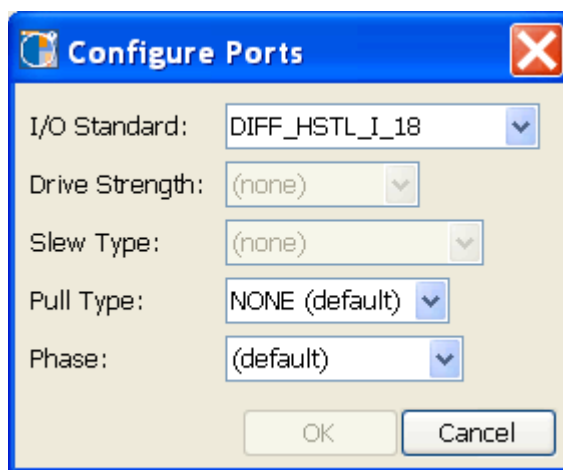


Figure 5-16: **Configure Ports dialog box**

3. View and edit the definable options in the Configure I/O Ports dialog box:
 - **I/O Standard**—Select the desired I/O Standard constraint.
 - **Drive Strength**—Select the desired Drive Strength value.
 - **Slew Type**—Select the desired Slew Type value.
 - **Pull Type**—Select the desired Pull Type value.
 - **Phase**—Enter a phase group or select an existing phase group. A phase group is a logical grouping of ports that is used in SSN calculations to indicate that the set of ports share the same frequency and phase. For more information on using this option, see [“Defining the I/O Port Switching Phase Groups.”](#)

Refer to Xilinx device documentation for information regarding voltage capabilities of the device.

Setting I/O Port Direction

Select the I/O ports, busses or interfaces to be configured, and select the **Set Direction** popup menu command in the I/O Ports view.

Defining Differential Pairs

To define a differential pin pair, select any two I/O ports, and select the **Make Diff Pair** popup menu command in the I/O Ports view.

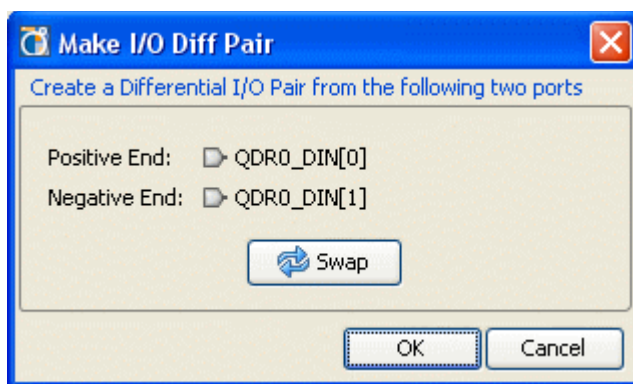


Figure 5-17: **Make I/O Diff Pair**

The two I/O Ports display in the dialog box with initial Positive End and Negative End definitions. Click the **Swap** button to alternate the Positive End and Negative End definitions.

Select the **Split Diff Pair** popup menu command to remove the differential pair definition on any differential pin pair.

Prohibiting I/O Pins and I/O Banks

PinAhead provides an Interface to selectively prohibit individual I/O pins, groups of I/O pins or I/O banks. You can select and prohibit pins in the Package Pins, Device, and Package views.

To prohibit I/O pins or I/O banks:

1. Select the I/O pins or I/O banks in the Package Pins or Package views.
2. Select **Set Prohibit** from the popup menu.

A red X symbol is placed on the prohibited pins in the Package view, and a checkmark is placed in the Prohibit column of the Package Pins view as shown in

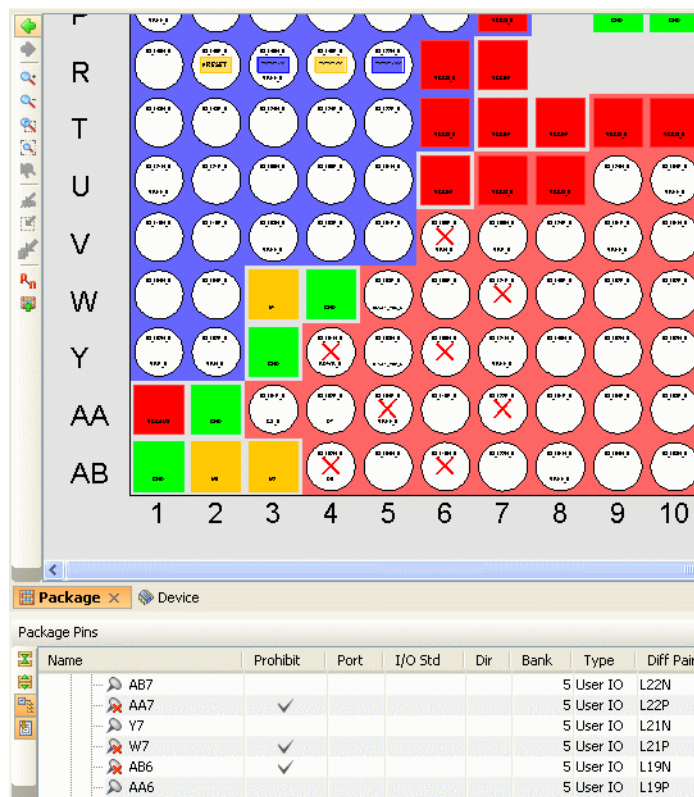


Figure 5-18: Setting Prohibits on Package Pins

Creating I/O Port Interfaces

Multiple ports or busses can be grouped together by creating an Interface. This aids in pin assignment by treating all of the interface ports as one group. Assigning all of the pins simultaneously helps condense and isolate the interface for clock region or PCB routing concerns. It also makes it much easier to visualize and manage all of the signals associated with a particular logic interface.

To create an Interface:

1. Select the desired signals in the I/O Ports view.
2. Select **Create I/O Port Interface** from the popup menu.

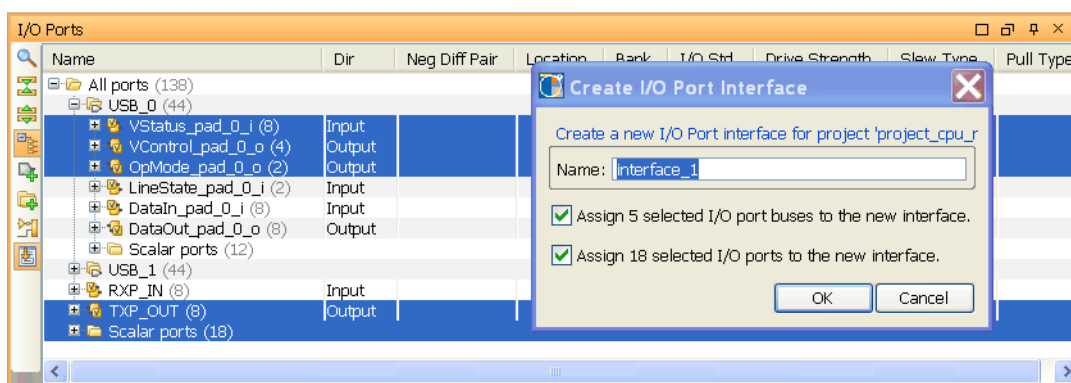


Figure 5-19: Create I/O Ports Interface

3. Enter a name for the Interface and adjust assignment selection.
4. Click **OK**.

The Interfaces appear as expandable folders in the I/O Ports view. Additional I/O ports can be added to the Interface by selecting them in the I/O Ports view and dragging them into the Interface folder.

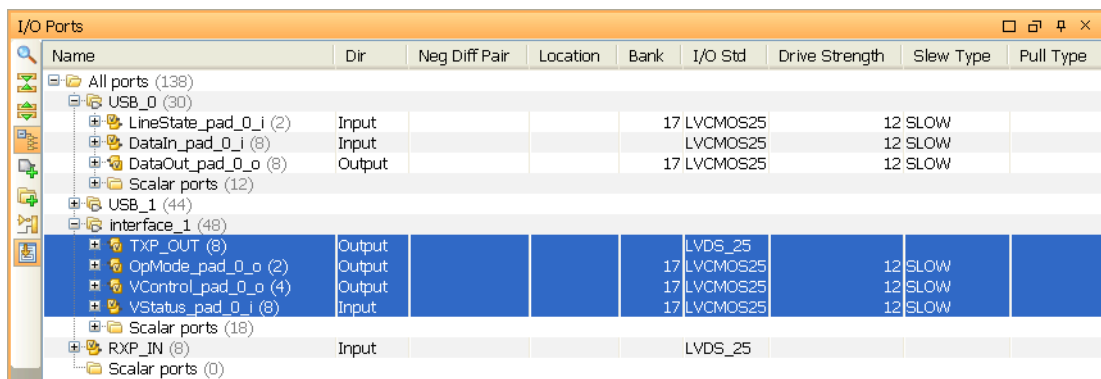


Figure 5-20: Manage I/O Port Interfaces

To include additional I/O ports in an Interface:

1. Select a port or bus.
2. Select **Assign to Interface** from the popup menu.
3. Select the target interface to which you will add the I/O ports.

To remove I/O Ports and Interfaces:

1. Select a port or Interface.
2. Select **Unassign from Interface** from the popup menu.

To delete Interfaces, select an Interface, and select **Delete** from the popup menu, or press the **Delete** key.

Placing I/O Ports

PinAhead provides a variety of ways to assign I/O Ports to package pins. Individual I/O Ports, groups of I/O Ports or Interfaces can be selected in the I/O Ports view and graphically assigned to package pins in the Package view or I/O pads in the Device view. Online DRCs can be toggled on or off during interactive placement.

The three placement mode options available for assigning I/O Ports are the following. Each is described below.

- Place I/O Ports in an I/O Bank
- Place I/O Ports in an Area
- Place I/O Ports Sequentially

For information on automatic I/O Port assignment, see [“Automatically Assigning I/O Ports,”](#) page 146.

Enabling Interactive Design Rule Checking

While the PlanAhead software is not a sign-off pin assignment tool, it does try to ensure a legal pin out. The interactive I/O placement routines check many common error cases. This capability is toggled on and off using the I/O Placement section of the General dialog box (accessible using **Tools > Options > General**). Using the feature will not allow placement of I/O Ports on pins that will cause a design issue. In Place I/O Ports Sequentially mode, when attempts are made to place an I/O Port on a problematic pin, a tool tip displays describing why the I/O Port is not able to be placed. The online DRC checks are enabled by default. Many of these checks can only run when a netlist representing the final design is loaded.

The interactive I/O placement rules include:

- Prohibiting placement on noise sensitive pins associated with Gigabit Transceivers “GTs”. Currently, all I/O potentially noise sensitive package pins are prohibited
- Prohibiting I/O standard violations
- Ensures that I/O standards are not used in banks that do not support them
- Ensuring that banks do not have incompatible VCC ports assigned
- Ensuring that banks that need a VREF ports have free VREF pins
- Proper assignment of global clocks and regional clocks (only with EDIF/NGC netlist and UCF imported)
- Ensuring Input and High Drive outputs only go to capable pins (for Spartan®-3 devices)
- Differential I/O ports to the proper sense pin
- Ensures that no output pins are placed on input-only pins

Placing I/O Ports into I/O Banks

To place I/O ports into I/O banks:

1. In the I/O Ports view, select an individual I/O port, a group of I/O ports or Interfaces.
2. Use one of the following commands:
 - Select **Place I/O Ports in an I/O Bank** from popup menu in the I/O Ports view.
 - Click the **Place I/O Ports in an I/O Bank** button in either the Package or Device view.



Figure 5-21: Place I/O Ports in an I/O Bank Button

The group of I/O ports is attached to the cursor when it is dragged over a package pin or I/O pad. A tool tip displays how many pins can be placed in the selected I/O bank.

3. Click on a pin or pad to assign the selected I/O ports.

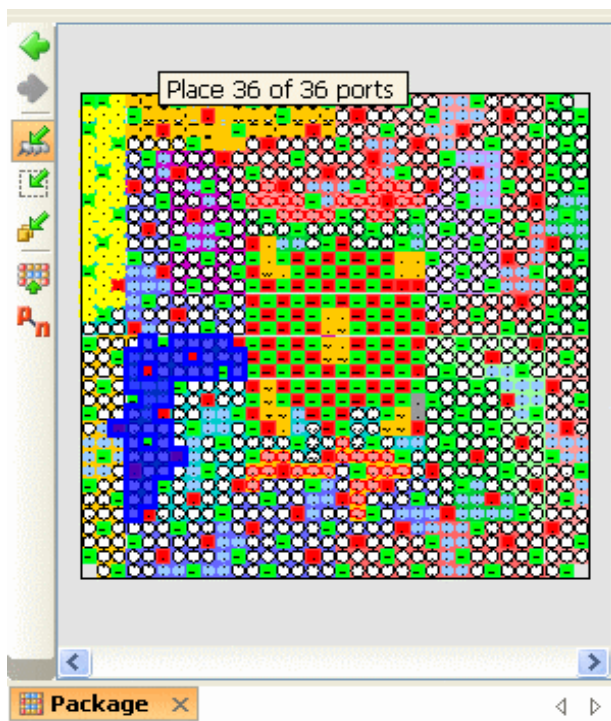


Figure 5-22: Placing I/O Ports in I/O Banks

4. If more I/O ports are selected than will fit in the I/O bank, the command is continued. The cursor will drag the remaining I/O ports to the next I/O bank selected, and so on until all of the I/O ports are placed, or until you press the **Esc** key.

Moving the cursor within the Package view will actively display the I/O pin coordinates on the top and left sides of the view. Additional I/O pin and bank information is displayed in the Status bar located at the bottom of the PlanAhead Environment. The active object being reported is highlighted in the Package view.

Holding the cursor over the Package view will invoke a tool tip that displays the pin information.

Ports are assigned in the order they appear in the I/O Ports view. The assignment order can be adjusted by applying sorting techniques in the I/O Ports view prior to assignment. The assignment order is also driven from the initial Pin that is selected for I/O bank assignment. Selecting a pin at one end of an I/O Bank will result in a continuous bus assignment across the I/O bank.

PCB routing concerns for busses are being considered. Pin ordering during assignment attempts to keep the bus bits vectored within the assignment area. Assignment patterns can be customized to address bus routing concerns.

Placing I/O Ports in a Defined Area

To place I/O Ports into a defined area:

1. In the I/O Ports view, select individual I/O ports, groups of I/O ports or Interfaces.
2. Use one of the following commands:
 - Select **Place I/O Ports in an Area** from the popup menu in the I/O Ports view.
 - Click the **Place I/O Ports in an Area** button in either the Package or Device view.



Figure 5-23: Place I/O Ports in an Area Button

The cursor turns into a cross symbol which indicates that you can define a rectangle for port placement.

3. Draw a rectangle in either the Package or Device view to define the assignment area.

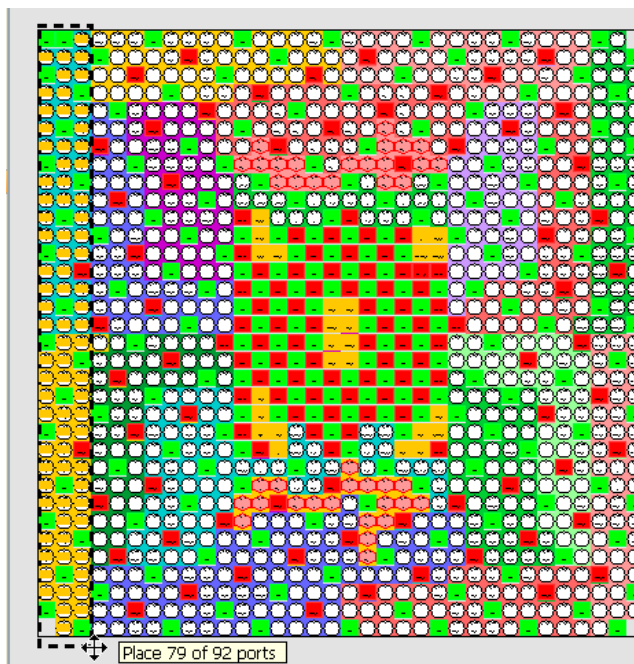


Figure 5-24: Placing I/O Ports in an Area

4. If more I/O Ports are selected than will fit in the area defined, the command is continued. The cursor will continue to display as a cross to draw another area to place the remaining I/O Ports, and so on until all of the I/O Ports are placed, or until you press the **Esc** key.

Ports are assigned in the order that they appear in the I/O Ports view. The assignment order can be adjusted by applying sorting techniques in the I/O Ports view prior to assignment.

The direction that the rectangle is drawn dictates the I/O ports assignment order. The I/O Ports is assigned in order from the inside pin of the first rectangle coordinate selected. Creative definition of the area rectangles can provide very useful pinout configurations from a PCB routing perspective.

Sequentially Placing I/O Ports

To place I/O ports sequentially:

1. In the I/O Ports view, select an individual I/O port, a group of I/O ports or Interfaces.
2. Use one of the following commands:
 - Select **Place I/O Ports Sequentially** from the popup menu in the I/O Ports view.
 - Click the **Place I/O Ports Sequentially** button in either the Package or Device view.



Figure 5-25: Place I/O Ports Sequentially Button

The first I/O port in the group is attached to the cursor when it is moved over a package pin or I/O pad. A tool tip displays the I/O port and package pin names.

3. Clicking on a pin or pad to assign an I/O port.



Figure 5-26: Placing I/O Ports Sequentially

4. If more I/O ports are selected, the command is continued. The cursor will drag the next I/O ports, and so on until all of the I/O ports are placed or until you press the **Esc** key.

Ports are assigned in the order that they appear in the I/O Ports view. The assignment order can be adjusted by applying sorting techniques in the I/O Ports view prior to assignment.

Automatically Assigning I/O Ports

PinAhead has the capability to automatically assign all or any selected I/O ports to package pins. The autoplacer will obey all I/O standard and differential pair rules, and will place global clock pins appropriately.

To automatically assign I/O ports to a subset of unassigned I/O ports:

1. Select the unassigned I/O ports in the I/O Ports view.
2. Select **Tools > Autoplace I/O Ports**, or in the I/O Ports view, select **Autoplace I/O Ports** from the popup menu.

The Autoplace I/O Ports wizard displays.

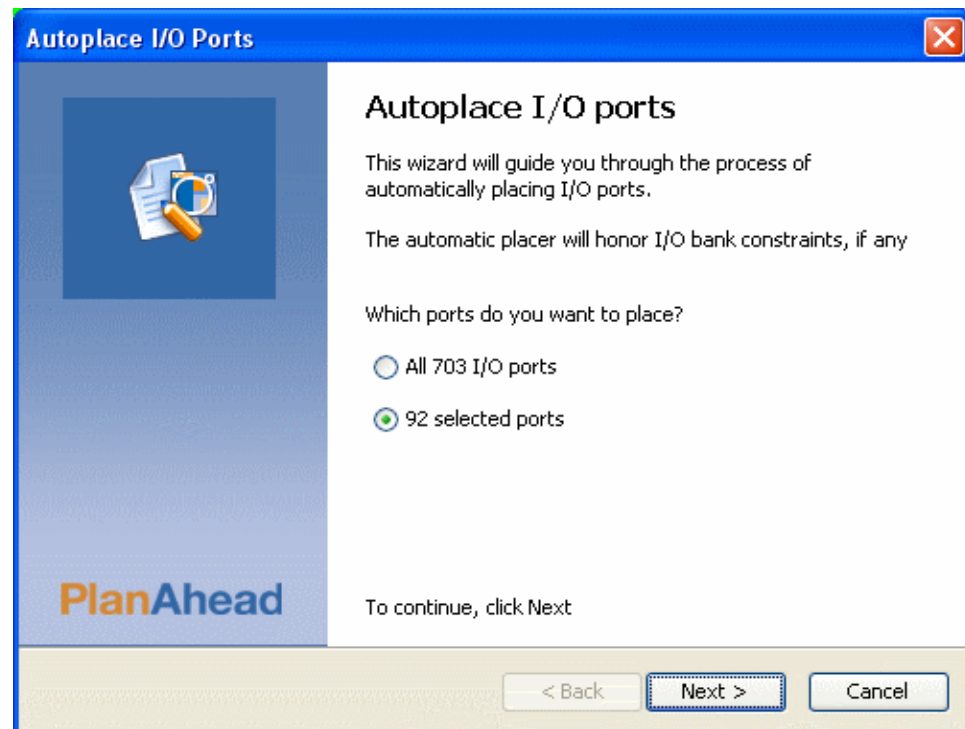


Figure 5-27: Autoplace I/O Ports Wizard

3. Select the group of I/O ports to place, and click **Next**.

If you select I/O ports that have already been assigned to package pins, the following dialog box opens.

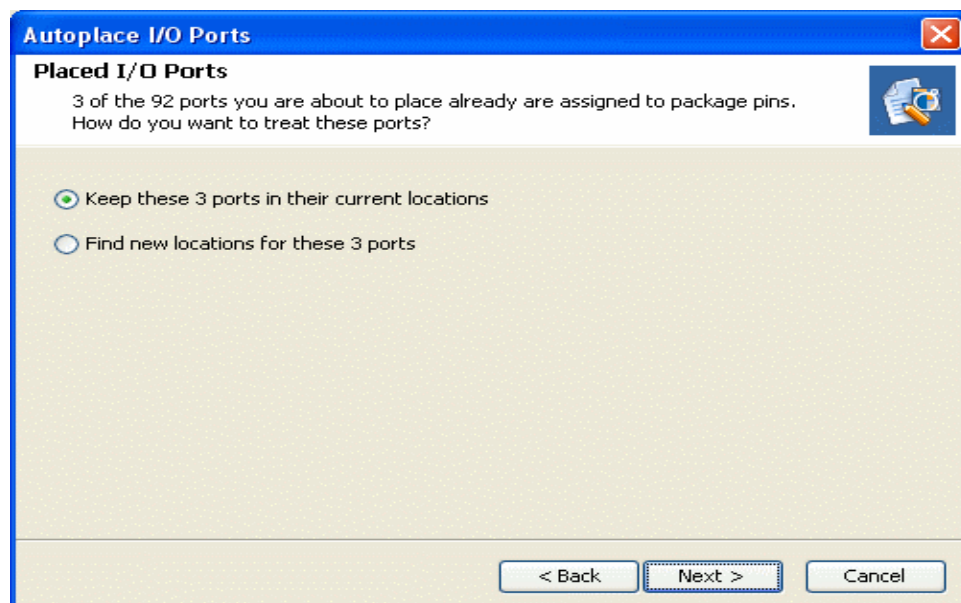


Figure 5-28: Autoplace I/O Ports Wizard

4. Select the I/O ports to place, and click **Next**.
5. Click **Finish** in the Summary page to place the selected I/O ports.

Placing Gigabit Transceiver I/O Ports

To better manage Gigabit Transceivers (GTs), PinAhead groups the two related I/O diff pairs and the GT logic object automatically during selection, placement, and moving. The GT objects get selected as one object and move together, which prohibits illegal assignment of the GT resources.

The noise sensitive I/O pins surrounding the GTs are prohibited automatically during port placement if the online DRCs are enabled. Refer to [“Enabling Interactive Design Rule Checking.”](#)

Placing I/O Related Clock Logic

Global and regional clock related logic, such as BUFGs, DCMs, BUFRs, DelayCtrls, etc., can be manually placed in the Device view. Appropriate logic sites are displayed for all device-specific resources. Select **Edit > Find** to search for various Site types available in the Device. They can then be selected in the Find Results view to highlight their location.

PlanAhead has many means to select the design clock related logic, such as the Find command or the Schematic and Netlist views.

To place clock logic manually:

1. Zoom in the Device view to locate the appropriate device site to place the logic.
2. Select the **Create Site Constraint Mode** toolbar icon.
3. Select the desired logic to place in the Find Results, Schematic or Netlist views, and drag it onto the available corresponding Device view site.

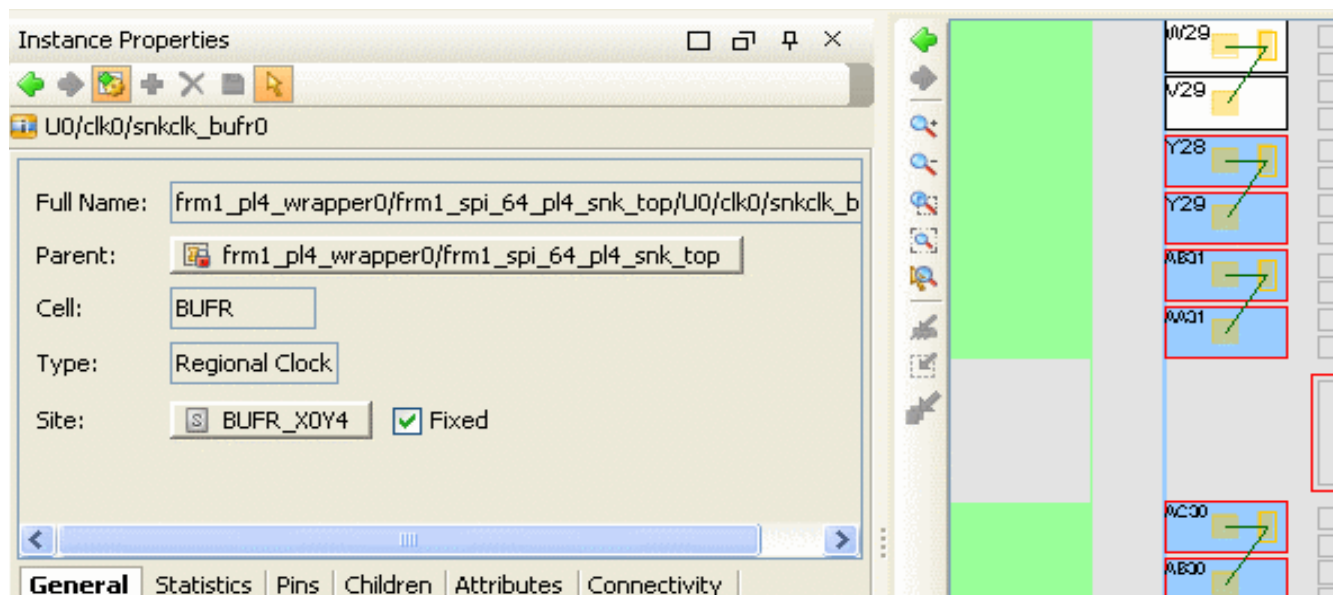


Figure 5-29: Manually Placing Clock Logic

Removing I/O Placement Constraints

Placement constraints can be removed by first selecting them, and then selecting the **Unplace** popup menu command.

Refer to [“Working with Placement LOC Constraints”](#) for information on selectively removing I/O related placement constraints.

Configuring DCI_CASCADE Constraints

The DCI_CASCADE constraint can be configured within the PlanAhead environment. For information about the intent and use of the DCI_CASCADE constraint, refer to the Xilinx *Constraints Guide*.

The basic premise of the constraint is to link two or more adjacent I/O Banks together for clocking purposes. The I/O bank with the input clock is called the master and all others are slaves. The constraint can be set by preselecting the desired I/O Banks prior to running the command or by selecting them within the command dialog box.

To configure the DCI_CASCADE constraint:

1. Optionally, select the desired I/O Banks to configure.
2. Right-click and select the **Create a DCI Cascade** popup menu command.

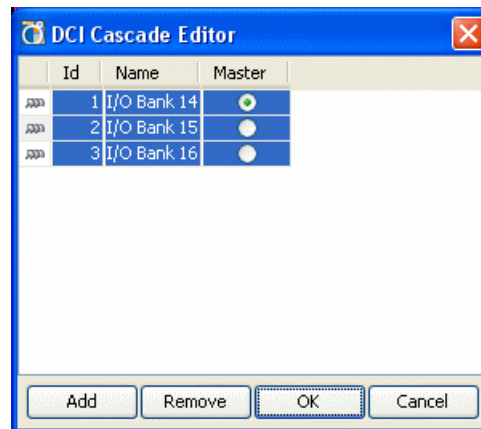


Figure 5-30: Creating a DCI Cascade

Preselected I/O Banks appear in the dialog box as shown in [Figure 5-31, page 150](#). Select the **Add** button to include additional I/O banks.

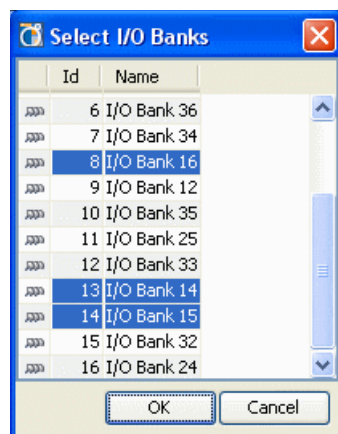


Figure 5-31: Selecting I/O Banks for the DCI Cascade

3. Select an I/O Bank to be the Master.
4. Click **OK**.

Notice that as I/O Banks are selected, they are highlighted in the other PlanAhead views.

The DCI Cascades are displayed in the Physical Hierarchy view as shown in Figure 5-32.

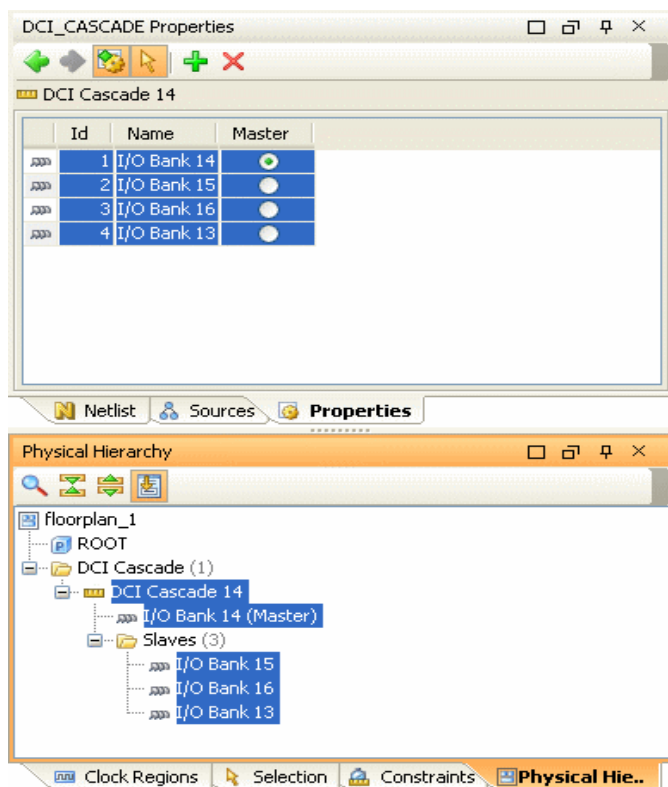


Figure 5-32: Viewing DCI Cascades

Editing a DCI Cascade

You can modify DCI Cascades by selecting the desired DCI Cascade in the Physical Hierarchy view and using the DCI_CASCADE Properties view.

To save all changes, click the **Apply** button in the DCI_CASCADE Properties view.

The Master can be changed by selecting a new master I/O Bank.

I/O banks can be removed from the DCI Cascade by selecting the I/O banks in the DCI_CASCADE Properties view, and clicking the **Delete I/O Banks** button.

Additional I/O banks can be included in the DCI Cascade by selecting them in the DCI_CASCADE Properties view and clicking the **Add I/O Banks** button. The Add I/O Banks dialog box is displayed allowing new I/O banks to be selected. The newly selected I/O banks are highlighted in the other PlanAhead views.

DCI Cascade constraints can be removed by selecting the constraint in the Physical Hierarchy view and using the **Delete** command.

Running I/O Port and Clock Logic Related DRCs

Running DRCs

This section details the steps involved in running I/O Port and clock-related DRCs. See [“Running Netlist and Floorplan DRCs” in Chapter 8](#) for information on running netlist and floorplan related DRCs.

Individual rules can be selected and run as follows:

1. Select **Tools > Run DRC**.

The Run DRC dialog box opens as shown in [Figure 5-33](#).

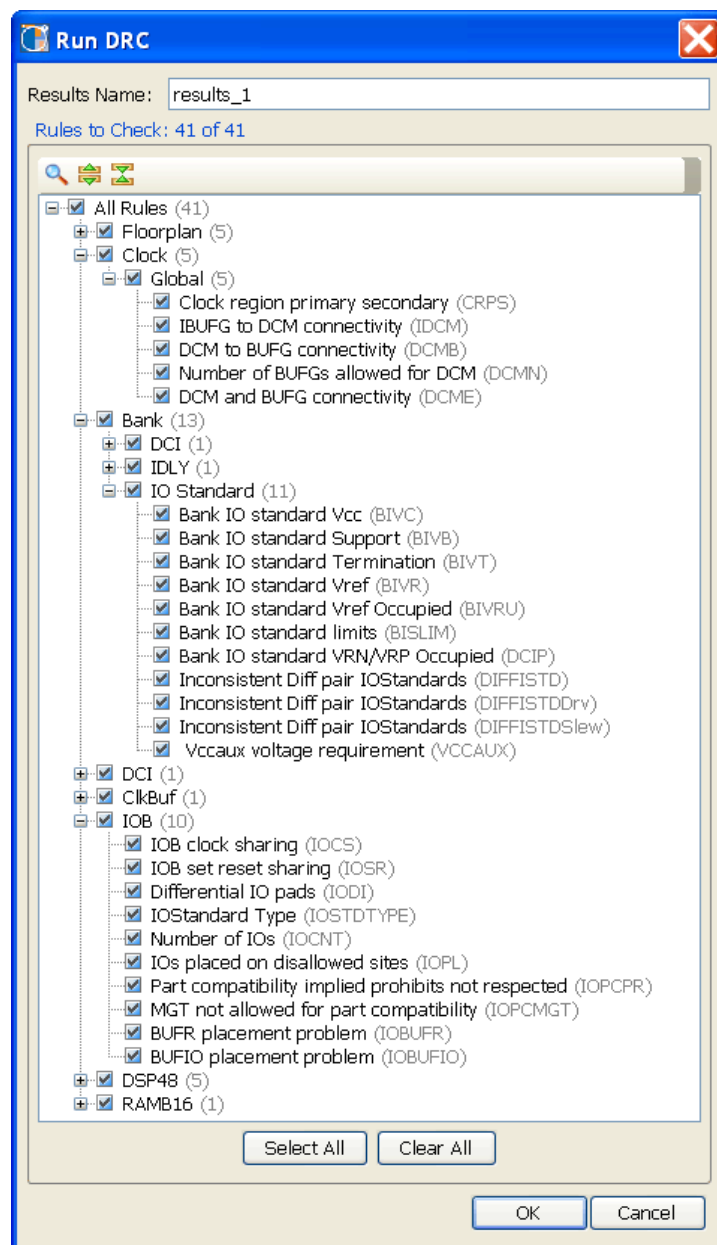


Figure 5-33: Run DRC Dialog Box: I/O Pin and Clock DRC Rules

2. View or edit the Results Name field. Enter a name for the results for a particular run for easier identification during debug in the DRC Violations browser. The output file name will match the name entered.
3. In the Rules to Check group box, use the check boxes to select the design rules to check for each design object. For information about each rule, see [“I/O Port and Clock Logic DRC Rule Descriptions.”](#)
 - Expand the hierarchy using the **Expand All** toolbar buttons, or click the + next to each category or design object.
 - Click the check box next to design object if you want all DRCs to be run, click individual DRCs, or click **All Rules** to run a complete DRC (all rules for all design objects)

- Click **OK** to invoke the selected DRC checks.

Viewing DRC Violations

Once completed, the DRC Results view opens.

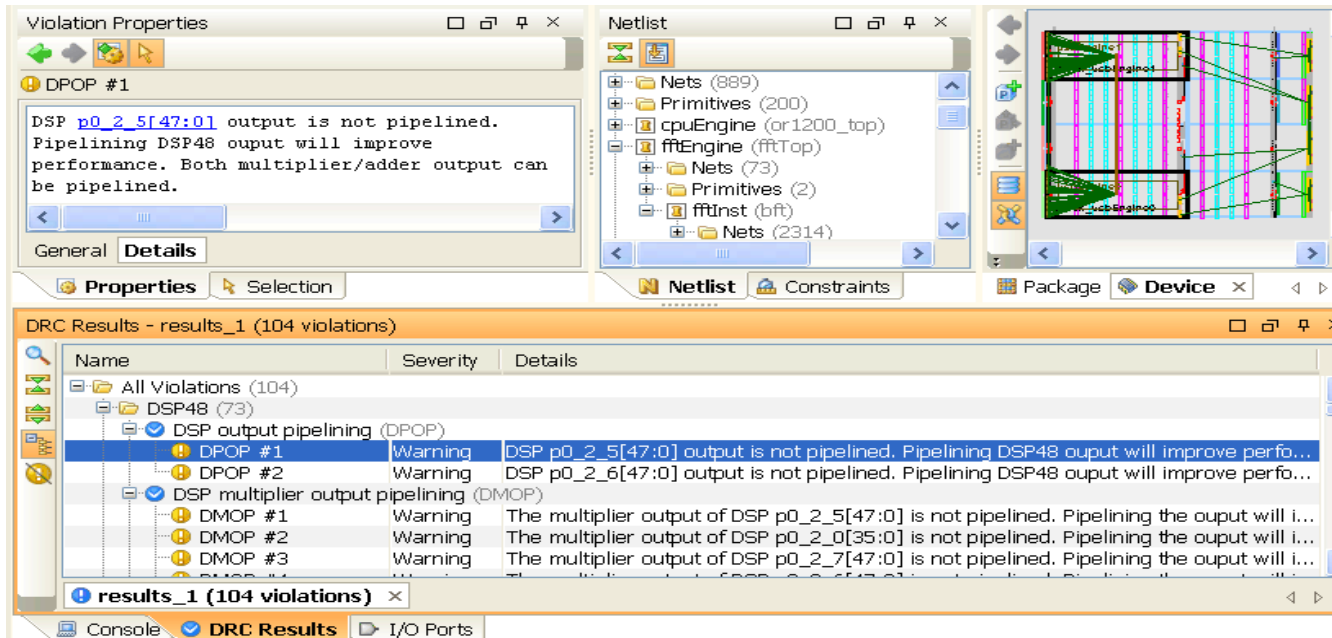


Figure 5-34: DRC Results

Each violation is expanded in the DRC Results view. Errors display a red icon. Warnings display an amber icon. Informational messages display a yellow icon. By default all errors and warnings are displayed. Click the Hide Warning and Information Messages toolbar button to hide all warnings and info messages, and view only errors. Click the toolbar button again to view all errors and warnings once again.



Figure 5-35: Hide Warning and Information Messages Button

Select an error in the DRC Results view list, and the specific violation information is displayed in the Properties view. Select a blue link in the Properties view to highlight the violating design elements in the Device view, Netlist view and Schematic view.

Violations will no longer be displayed in the DRC Results view after the error condition is rectified and DRC is rerun.

Each time the Run DRC command is run and errors are detected, a new results tab is added to the DRC Results view. A separate results output file is also created in the PlanAhead invocation directory.

I/O Port and Clock Logic DRC Rule Descriptions

The following tables describe the various DRC rules, rule intent and severity.

Note: Other DRC rule descriptions are found in “DRC Rule Descriptions” in Chapter 8.

- “Global Clock Rules”
- “IOB Rules”
- “Bank I/O Standard Rules”

Note: The I/O Port and Clock Logic DRCs available in PlanAhead is not an exhaustive list of I/O-related DRCs. Consult the device documentation for more information on I/O ports and clock region specifications.

Global Clock Rules

Table 5-1: Global Clock Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Clock Region Primary Secondary	CRPS	Only one of the primary/secondary pair clocks have access to any one quadrant via global clock routing resources. If these two clocks drive clock inputs in the same quadrant, the nets will not be routable using the global clock routing resources. (Spartan-3 devices) Note: If the Pblock spans more than one clock region, the error reported may not be an issue for the ISE® software to place the logic in the appropriate clock regions with the Pblock.	Error
IBUFG to DCM connectivity	IDCM	IBUFGs have dedicated routing only to all DCMs on the same edge (top, bottom, left, right) of the device.	Warning
DCM to BUFG connectivity	DCMB	DCM can connect to a maximum of four BUFGs. There are pairs of buffers with shared dedicated routing resources such that if both are driven by the same DCM, one of the two will necessarily be driven using non-dedicated routing resources; this causes the design to fail. If the buffers are numbered 1 through 8 from left to right, there are four pairs of exclusives: 1:5, 2:6, 3:7, 4:8. If a buffer is placed in Site 1, another driven by the same DCM should not be placed in Site 5.	Error
Number of BUFGs allowed for DCM	DCMN	DCM can connect only up to 4 BUFGs. This is related to DCMB.	Error
DCM and BUFG connectivity	DCME	BUFGs have dedicated routing only to all DCMs on the same edge (top, bottom, left, right) of the device.	Warning

IOB Rules

Table 5-2: IOB Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
IOB clock sharing	IOCS	IOB sites are divided into pairs for the purpose of sharing clock routing resources. These pairs are normally LVDS pairs as well. In some cases there could be routing issues based on how the flops are packed inside the IOB. To resolve this issue flops need to be assigned to specific BEL.	Warning
IOB set reset sharing	IOSR	IOB site has input, output and tri-state registers, each of these registers share same set/reset signal. Will not be able pack registers with different reset signals.	Error
Differential I/O pads	IODI	Differential I/O P and N signals should be LOCed in dedicated differential pair.	Error
I/O Standard Type	IOSTDTYPE	Ensures that a Diff pair I/O Standard has been applied to diff pair pins only.	Warning
Number of IOs	IOCNT	Indicates whether more I/O Ports are defined than there are pins in the target device.	Warning
Non inputs placed on input only pins	IOPR	Checks that a port is not placed on a prohibited pin.	Error
Diff term loced to low capacitance IOB Site	IOLVDSCC	Checks that differential output standards are not used on low-capacitance sites that cannot support them.	Error
Prohibit not specified for part compatibility	IOPCPR	For designs that use part compatibility, checks that if any package pin does not exist on at least one compatible part, it is marked as "prohibit" and nothing is placed on it.	Error
MGT not allowed for part compatibility	IOPCMGT	Indicates whether part compatibility is used with two parts that have different MGT supply voltages, thereby disallowing the use of MGT.	Warning
Regional Clock Term has no BUFR site	IOBUFR	Checks that a regional clock terminal and the related BUFR are placed at mutually routable locations.	Error
Regional Clock Term has no BUFIO site	IOBUFIO	Checks that a regional clock terminal and the related BUFIO are placed at mutually routable locations.	Error

Bank I/O Standard Rules

Table 5-3: Bank I/O Standard Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Bank I/O Standard Vcc	BIVC	IOSTANDARD based VOUT voltage compatibility check for IOs in that bank.	Error
Bank I/O Standard Support	BIVB	Checks that the I/O Standard is supported in the I/O bank.	Error
Bank I/O standard Termination	BIVT	IOSTANDARD based DCI Termination voltage compatibility check for IOs in that bank.	Error
Bank I/O Standard VREF	BIVR	IOSTANDARD based VREF voltage compatibility check for IOs in that bank.	Error
Bank I/O Standard VREF Occupied	BIVRU	IOSTANDARD based VREF voltage compatibility check for IOs in that bank.	Error
Bank I/O Standard VRN/VRP Occupied	DCIP	There are dedicated VRP and VRN I/O sites in I/O bank which can also be used as regular IOs. If a DCI I/O standard is used in that bank these IOs should be left unoccupied.	Error
Bank I/O Simultaneous Switching Output Limits	BISLIM	Checks the I/O placed within an I/O bank against Simultaneous Switching Output Noise.	Error
Inconsistent Diff pair I/O Standards	DIFFISTD	Checks that the terminals of a differential pair have the same I/O standard.	Error
Inconsistent Diff pair I/O Standards	DIFFISTDDrv	Checks that the terminals of a differential pair have the same drive.	Error
Inconsistent Diff pair I/O Standards	DIFFISTDSlew	Checks that the terminals of a differential pair have the same slew.	Error
Vccaux Voltage requirement	VCCAUX	Warns of any requirements on Vccaux, based on used I/O standards.	Warning

Running Simultaneous Switching Noise (SSN) Analysis

Running SSN Analysis (Virtex-6)

PlanAhead now incorporates the new Xilinx Simultaneous Switching Noise (SSN) predictor to improve the prediction of simultaneous switching output noise in Virtex®-6 devices. This tool is designed to provide estimates of the disruption that switching outputs will cause on other outputs in the I/O bank. The SSN Predictor incorporates I/O bank-specific electrical characteristics into the prediction to better model package effects on SSN. Since all power distribution networks within a packaged FPGA have different responses to noise, it is relevant to understand not only the I/O standards and number of I/O in a design, but also the response of the device power system to this switching.

In the Virtex-6 family of devices, I/Os are grouped into separate isolated I/O Banks, each with its own unique power distribution networks. These each have unique responses to switching activity. Xilinx characterizes all banks in the Virtex-6 family through three-dimensional extraction and simulation. This information is incorporated into the SSN predictor such that it can take the expected switching profile of a device and predict how the switching will affect the system's power network, and in turn how other outputs in the I/O bank are affected.

The SSN predictor is the best method available for accurately predicting how output switching will affect interface noise margins. The calculation and results are based on a wide range of variables. These estimates are intended to identify potential noise-related issues in your design and should not be used as final design "sign off" criteria.

PlanAhead uses the SSN calculation if a Virtex-6 part is selected. If using a device other than Virtex-6, refer to ["Running Weighted Average Simultaneous Switching Output \(WASSO\) Analysis."](#)

To run the SSN predictor:

1. Select **Tools > Run SSN Analysis**.

The Run SSN Analysis dialog box opens.

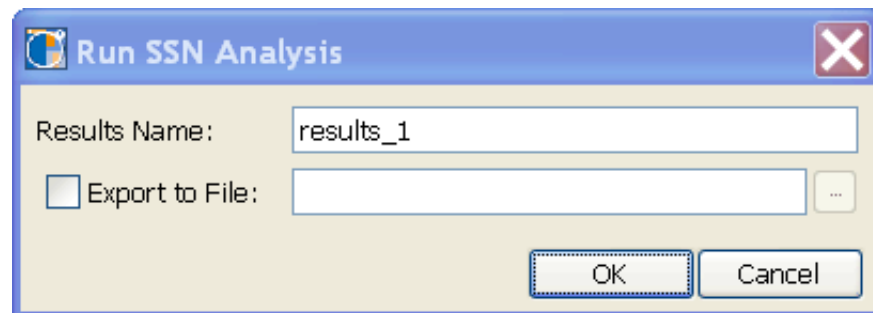
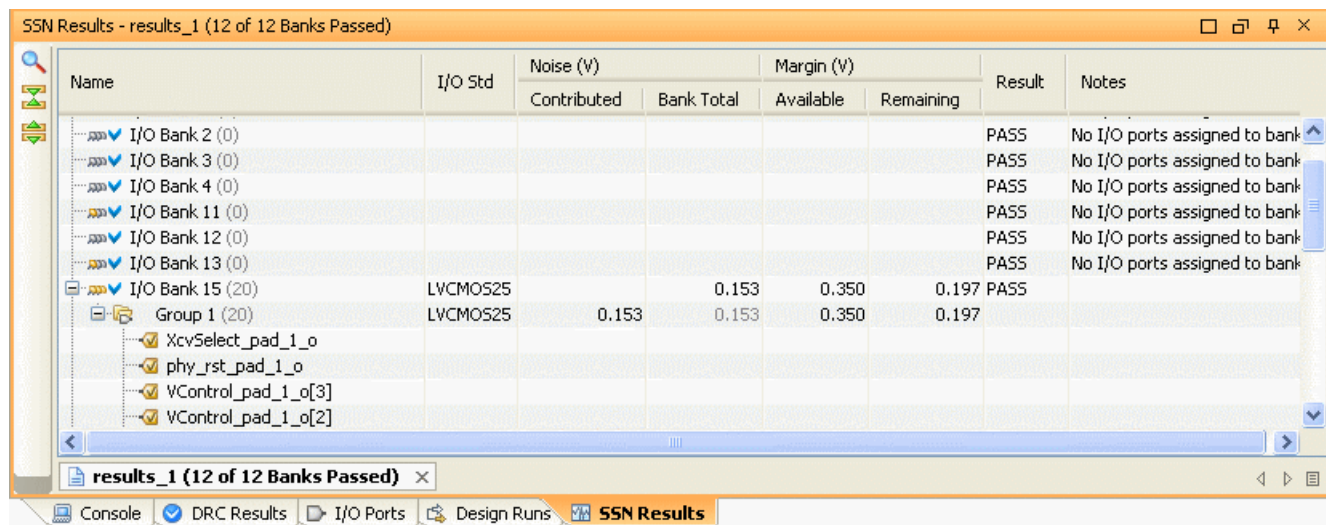


Figure 5-36: Run SSN Analysis Dialog Box

2. Optionally, enter a name in the Results Name field to identify the results in the SSN Results view.
3. Optionally, select **Export to File**, and enter an output file name in the Output File field and browse to select a location to write an external CSV format report file.
4. Click **OK**.

Viewing the SSN Results

The SSN Results view is shown in [Figure 5-37](#).



Name	I/O Std	Noise (V)		Margin (V)		Result	Notes
		Contributed	Bank Total	Available	Remaining		
I/O Bank 2 (0)						PASS	No I/O ports assigned to bank
I/O Bank 3 (0)						PASS	No I/O ports assigned to bank
I/O Bank 4 (0)						PASS	No I/O ports assigned to bank
I/O Bank 11 (0)						PASS	No I/O ports assigned to bank
I/O Bank 12 (0)						PASS	No I/O ports assigned to bank
I/O Bank 13 (0)						PASS	No I/O ports assigned to bank
I/O Bank 15 (20)	LVCMS25		0.153	0.350	0.197	PASS	
Group 1 (20)	LVCMS25	0.153	0.153	0.350	0.197		
XcvSelect_pad_1_o							
phy_rst_pad_1_o							
VControl_pad_1_o[3]							
VControl_pad_1_o[2]							

Figure 5-37: SSN Results View

The SSN Results view displays the following information:

- Name – Displays the I/O Banks available in the device. Each I/O bank has pin icons indicating how full the bank is, and a check mark or red circle indicating whether it passed or failed.
 - Group – Displays the various groups of pins with like I/O standards assigned within the bank, and displays their status. Groups are determined automatically based on the I/O standards, drive strength, slew rate, and phase assigned.
- Noise (V)
 - Contributed – Contains the SSN aggregate of each group, generated by the I/O standard, drive strength, and slew type of that group.
 - Bank Total – Defines aggregate SSN predicted for a bank or group. If multiple phases are specified for the groups of a given bank, the SSN contributions of groups with different phases will be accumulated separately, and the maximum of these will be reported. Because the SSN of an output are isolated to the output of that Bank, one SSN Bank total does not affect another Bank total. This column identifies which I/O groups are creating the most SSN, and how much margin they use up.
- Margin (V)
 - Available – Defines the allowable noise margin for that particular I/O standard on the high side of the signal as it switches to a 1. It is strictly based on the DC logic levels implicit in the I/O standard (no quantity information is taken into account) and represents the margin that the weakest drive of a high signal is above the JEDEC input thresholds. These margin values assume the weakest drive conditions, JEDEC/Spec termination, and standard receiver requirements for the standard. This is one place where conservative assumptions are made in the analysis, providing some guard band.
 - Remaining – Displays the amount of noise margin that is left over after accounting for all SSN in the bank.

Results are displayed in red when over 100% of the available margin is lost to SSN. The SSN predictor arrives at this value by subtracting the bank total predicted SSN (a per-bank number) from the noise margin of I/O standard (a per-group number).

- Result – Displays a Pass or Fail condition with failures displayed in red.
- Notes – Displays information about the I/O bank or groups.

The SSN results are relative to the state of the design when the SSN Analysis is run. It is not a dynamic report.

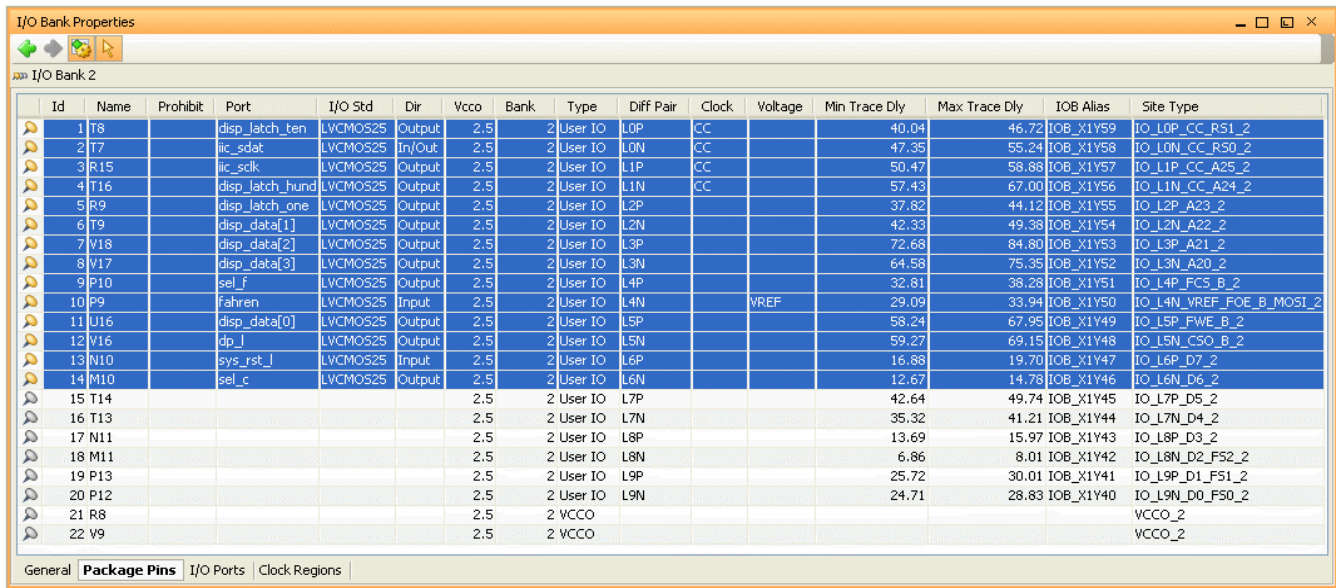
Resolving SSN Issues

When a violation occurs there are a number of ways to improve the results:

- The offending group(s) can be spread across multiple synchronous phases.
- If the Result is a Fail condition, assigning phase groups to ports that are switching concurrently. See [“Defining the I/O Port Switching Phase Groups.”](#)
- The offending group can be spread across multiple banks. This reduces the number of aggressive outputs on one bank’s power system.
- The offending group can use I/O standards that have a lower SSN impact. Changing to a lower drive strength, a parallel-terminated DCI I/O standard, or a lower class of driver can improve SSN, for example, changing SSTL Class II to SSTL Class I.
- The offending group can be phase shifted by 90 degrees if at DDR rate, or by 180 degrees if at SDR rate. This puts half the aggressive output switching out-of-phase with the other half, and instead of interfering constructively, it interferes destructively.
- If the Result is a Fail condition, assign phase groups to ports that are switching concurrently. See [“Defining the I/O Port Switching Phase Groups.”](#)

Viewing I/O Bank Properties

Selecting an I/O bank in the SSN Results view displays information about the I/O ports, pins and groups assigned to the I/O bank in the I/O Bank Properties view. Select the **General** tab to view information about the number and types of Ports assigned to the I/O Bank. Select the **Package Pins** or **I/O Ports** tab to view the detailed information about the Pins or Ports within the bank, as shown in Figure 5-38.



Id	Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	Type	Diff Pair	Clock	Voltage	Min Trace Dly	Max Trace Dly	IOB Alias	Site Type
1 T8	disp_latch_ten		disp_latch_ten	LVCNMOS25	Output	2.5	2	User IO	L0P	CC		40.04	46.72	IOB_X1Y59	IO_L0P_CC_RS1_2
2 T7	iic_sdat		iic_sdat	LVCNMOS25	In/Out	2.5	2	User IO	L0N	CC		47.35	55.24	IOB_X1Y58	IO_L0N_CC_RS0_2
3 R15	iic_sclk		iic_sclk	LVCNMOS25	Output	2.5	2	User IO	L1P	CC		50.47	58.88	IOB_X1Y57	IO_L1P_CC_A25_2
4 T16	disp_latch_hund		disp_latch_hund	LVCNMOS25	Output	2.5	2	User IO	L1N	CC		57.43	67.00	IOB_X1Y56	IO_L1N_CC_A24_2
5 R9	disp_latch_one		disp_latch_one	LVCNMOS25	Output	2.5	2	User IO	L2P			37.82	44.12	IOB_X1Y55	IO_L2P_A23_2
6 T9	disp_data[1]		disp_data[1]	LVCNMOS25	Output	2.5	2	User IO	L2N			42.33	49.38	IOB_X1Y54	IO_L2N_A22_2
7 V18	disp_data[2]		disp_data[2]	LVCNMOS25	Output	2.5	2	User IO	L3P			72.68	84.80	IOB_X1Y53	IO_L3P_A21_2
8 V17	disp_data[3]		disp_data[3]	LVCNMOS25	Output	2.5	2	User IO	L3N			64.58	75.35	IOB_X1Y52	IO_L3N_A20_2
9 P10	sel_f		sel_f	LVCNMOS25	Output	2.5	2	User IO	L4P			32.81	38.28	IOB_X1Y51	IO_L4P_FC5_B_2
10 P9	Fahren		Fahren	LVCNMOS25	Input	2.5	2	User IO	L4N		VREF	29.09	33.94	IOB_X1Y50	IO_L4N_VREF_FOE_B_MOSI_2
11 U16	disp_data[0]		disp_data[0]	LVCNMOS25	Output	2.5	2	User IO	L5P			58.24	67.95	IOB_X1Y49	IO_L5P_FWE_B_2
12 V16	dp_l		dp_l	LVCNMOS25	Output	2.5	2	User IO	L5N			59.27	69.15	IOB_X1Y48	IO_L5N_C50_B_2
13 N10	sys_rst_l		sys_rst_l	LVCNMOS25	Input	2.5	2	User IO	L6P			16.88	19.70	IOB_X1Y47	IO_L6P_D7_2
14 M10	sel_c		sel_c	LVCNMOS25	Output	2.5	2	User IO	L6N			12.67	14.78	IOB_X1Y46	IO_L6N_D6_2
15 T14						2.5	2	User IO	L7P			42.64	49.74	IOB_X1Y45	IO_L7P_D5_2
16 T13						2.5	2	User IO	L7N			35.32	41.21	IOB_X1Y44	IO_L7N_D4_2
17 N11						2.5	2	User IO	L8P			13.69	15.97	IOB_X1Y43	IO_L8P_D3_2
18 M11						2.5	2	User IO	L8N			6.86	8.01	IOB_X1Y42	IO_L8N_D2_FS2_2
19 P13						2.5	2	User IO	L9P			25.72	30.01	IOB_X1Y41	IO_L9P_D1_FS1_2
20 P12						2.5	2	User IO	L9N			24.71	28.83	IOB_X1Y40	IO_L9N_D0_FS0_2
21 R8						2.5	2	VCCO							VCCO_2
22 V9						2.5	2	VCCO							VCCO_2

Figure 5-38: I/O Bank Properties: Package Pins

Defining the I/O Port Switching Phase Groups

In some cases, different groups of I/O within a bank have a synchronous phase offset from one another, meaning it is not possible for them to switch simultaneously. This is true of data and strobe signals in many memory interfaces. In this case, proper SSN accounting must be informed by phase information. A phase group is a logical grouping of ports that are all in phase with each other from a timing perspective (i.e., their clocks have the same frequency and phase). By creating a grouping phase, not only is a group created, but I/Os with a different phase are being separated. The noise produced by the groups within a bank is summed to get the total noise for that bank, and if all outputs are either in phase with each other, or do not have a synchronous relationship, the output can be expected to switch (change values) at the same time. If in the SSN analysis a bank is failing, phase groups can be used to group ports that are at separate synchronous phases, thus lowering the total noise for that bank when the SSN predictor is run again.

To set the switching phase for a single or set of I/O Ports:

1. Select one or more I/O Port in any of the PinAhead views.
2. In the I/O Ports view, Package Pins view, or SSN view, select the **Configure I/O Ports** popup menu command as shown in Figure 5-39, page 161.

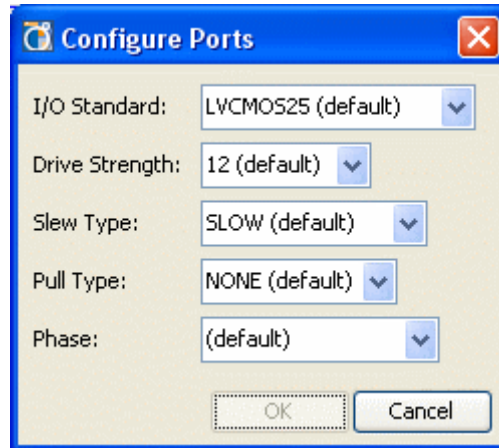


Figure 5-39: **Configure Ports dialog box**

3. In the Configure Ports dialog box, make sure the I/O Standard is correct.
4. If the port(s) are in phase, leave the Phase as default, or enter a unique phase, such as "phase 1".
5. Click **OK**.

Once the appropriate phase groups are assigned, rerun the SSN analysis.

Note: Asynchronous groups should not be treated as separate synchronous phases, as it is possible for them to switch simultaneously.

Running Weighted Average Simultaneous Switching Output (WASSO) Analysis

Running WASSO Analysis (Spartan-3, Virtex-4, Virtex-5)

PlanAhead contains a set of Weighted Average Simultaneous Switching Output (WASSO) checks to validate signal integrity of the device based on the I/O pin and bank assignments made in the design.

To run an analysis, select **Tools > Run WASSO Analysis**.

The Run WASSO Analysis dialog box opens, as shown in [Figure 5-40, page 162](#).

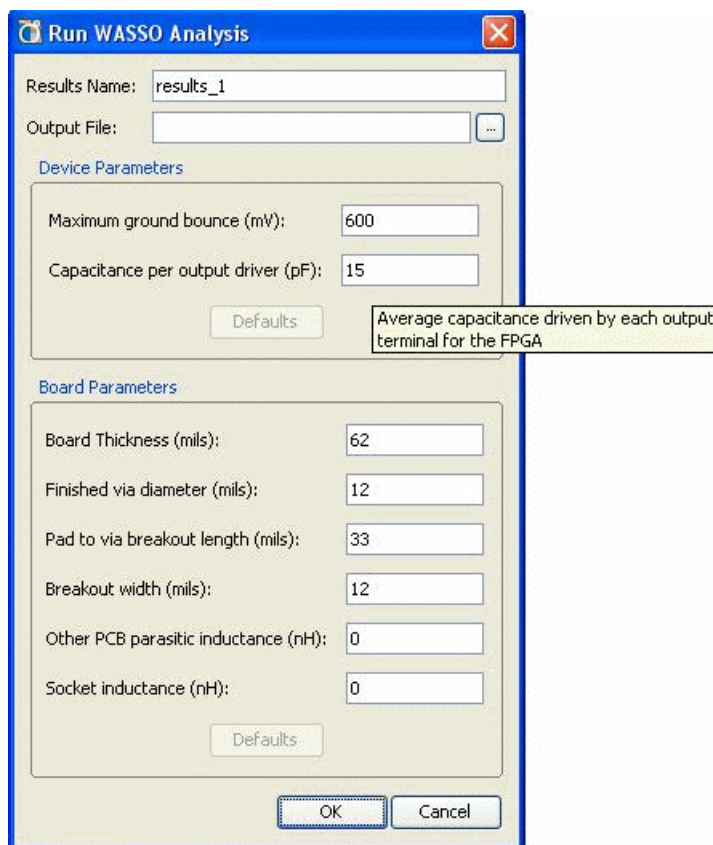


Figure 5-40: Run Wasso Analysis Dialog Box

The Output File field can be used to specify a report file name and location to write to disk.

Tool tips are provided when dragging the cursor over each of the entry fields indicating what values to enter.

Viewing the WASSO Analysis Results

The Device and Board Parameter values can be modified to reflect your specific design. The analysis is performed across the entire device first and then within each I/O bank as they relate to their neighboring I/O banks. The WASSO Results view displays in the Workspace as shown in [Figure 5-41, page 163](#).

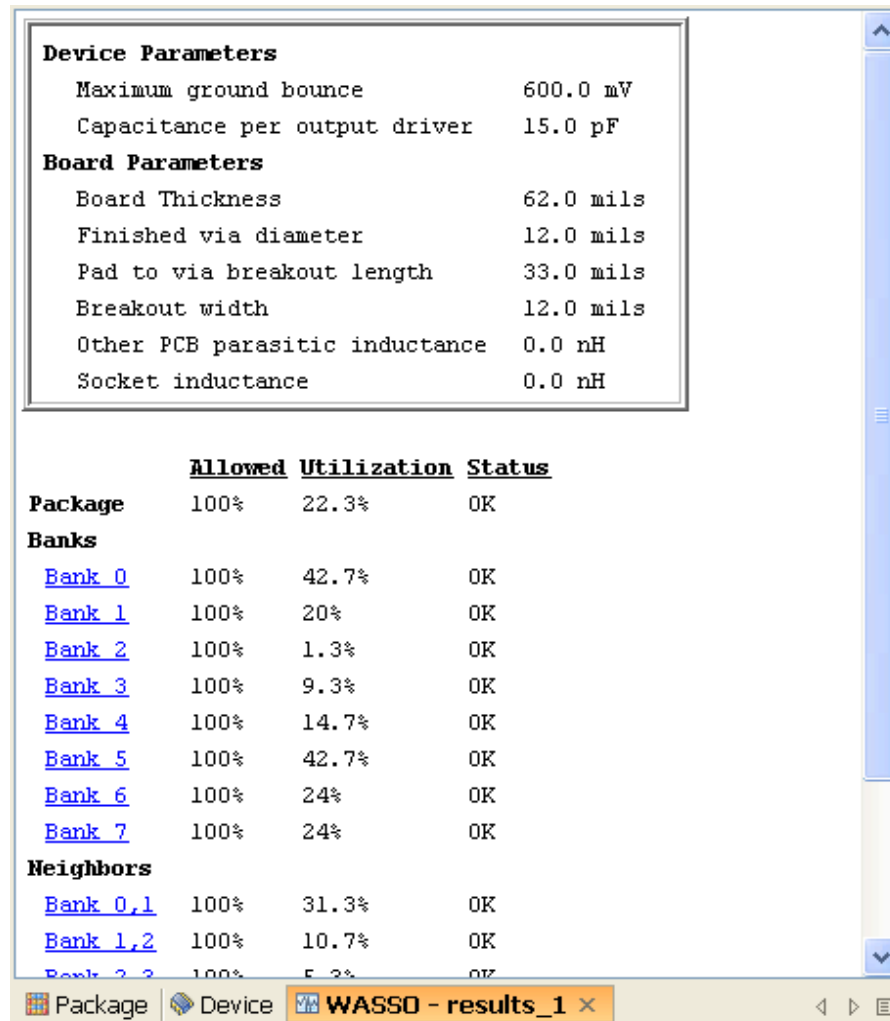


Figure 5-41: WASSO Results

Notice that the report lists allowable loading, utilization and status for I/O banks, and neighboring pair.

Exporting Package Pin Information

The device package pin information can be exported from PlanAhead to a CSV format file. The exported information includes information about all of the package pins in the device as well as design-specific I/O Port assignments and their configuration.

The package pin section of the exported list makes a great starting point for defining I/O Port definitions in a spreadsheet format.

Refer to the “[Importing I/O Ports](#)” section in this chapter for information on the exported CSV file format.

Exporting an I/O Port List

The I/O Port list can be exported from PlanAhead to an HDL, UCF, or CSV format file for use with RTL coding or PCB schematic symbol creation. To export the I/O Ports list information, select the **File > Export I/O Ports** command.

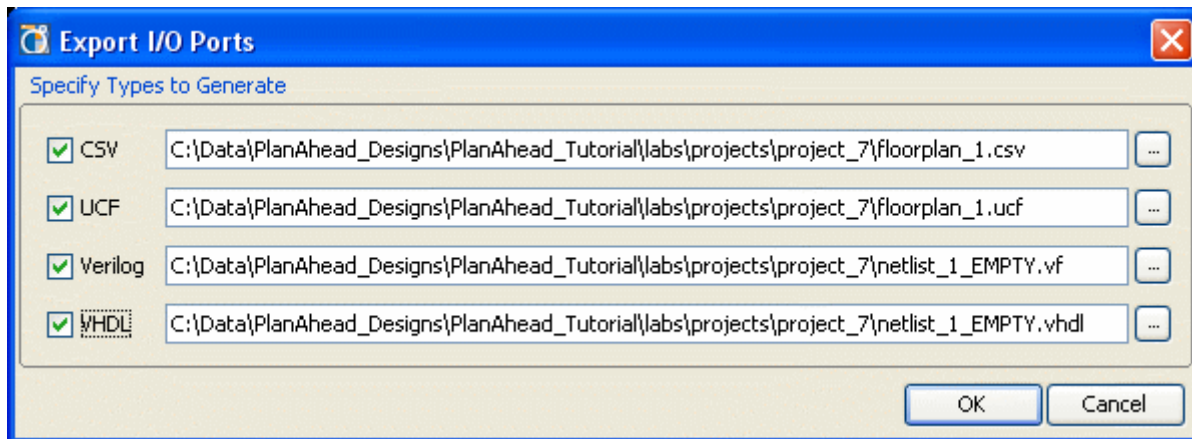


Figure 5-42: Exporting I/O Port Lists

Creating and Analyzing the RTL Design

The PlanAhead™ software Project environment enables you to create and manage RTL design files. Included in PlanAhead is basic source file management, an RTL Editor, an RTL schematic viewer, a set of RTL DRCs, and a resource estimator.

Logic synthesis and implementation can then be run using PlanAhead. For more information about running synthesis and implementation, refer to [Chapter 7, “Implementing a Design.”](#)

This chapter contains the following sections:

- [“Using the Project Environment”](#)
- [“Adding Sources to the Project”](#)
- [“Using the RTL Editor”](#)
- [“Elaborating and Analyzing the RTL Design”](#)

Using the Project Environment

The Project environment is used to import, develop and elaborate the RTL source files, and analyze the RTL design. Currently, Verilog, VHDL and core level NGC/NGO files can be imported and managed within the PlanAhead RTL Project.

Using the Sources View

The Sources view displays the RTL source directories and files imported into the Project. These files are either Verilog or VHDL format RTL sources, NGC/NGO core netlists, or XCF constraints files for XST synthesis

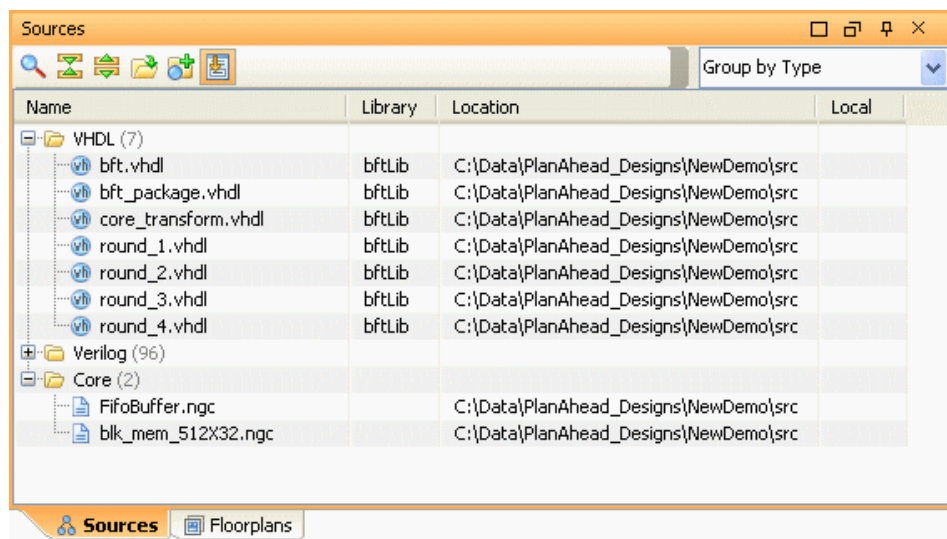


Figure 6-1: Sources View

The Sources view displays the source files. It also displays the source file type, VHDL library, location and whether the files are imported locally in the Project or referenced remotely.

- **Name**—List files alphabetically, or grouped by type of source: VHDL, Verilog, Core, SDC or XNF.
- **Library**—Displays the VHDL library defined for each file
- **Location**—Displays the location of the imported or externally references files
- **Local**—Indicates whether the files were imported into the Project or referenced remotely

Displaying Sources According to Category

The sources can be grouped according to category. The selection menu in the upper right corner of the Sources view is used to select the format to display the project sources.

- Select **Group by Type** to organize the source by VHDL, Verilog, Cores, SDC, or XNF.
- Select **Group by Source Root** to organize the list by source directory or location.
- Select **Flat View** to list all sources sorted alphabetically.

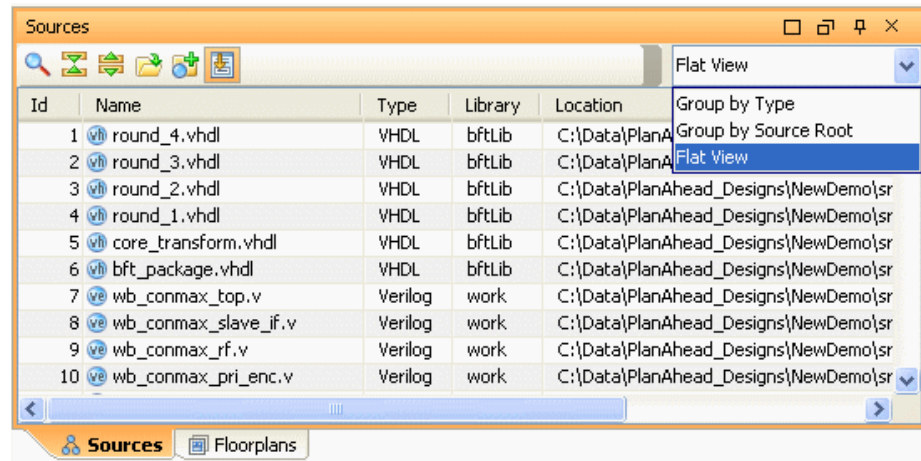


Figure 6-2: Defining the Display Format for Sources

Using the Sources View Specific Popup Menu Commands

Source files can be added, viewed or modified using the Sources view popup menu commands. The common popup menu commands are covered in the [“Using Common Popup Menu Commands,” page 251](#). The Sources view specific commands and a brief description of each are as follows:

- **Source File Properties**—Invokes the Source File Properties view.
- **Enable Source Files**—Sets the source file status to active for elaboration and synthesis. Source files can be toggled between Enabled and Disabled to define source file configurations.
- **Disable Source Files**—Sets the source file status to inactive for elaboration and synthesis. Source files can be toggled between Enabled and Disabled to define source file configurations. Disabled Source files appear in a shaded grey color to indicate disabled status.
- **Remove from Project**—Deletes the selected source files from the PlanAhead project. It also removes the files from the PlanAhead project disk location if the files were initially imported into the project.
- **Find in Files**—Invokes the *Find in Files* dialog box to enter text strings to search in the selected files. The Find in Files Results view is displayed with the results of the search. For more information, see [“Using the Find in Files Command to Search Source Files.”](#)
- **Open file**—Opens the selected file(s) in the RTL Editor view.
- **Add Sources**—Imports all of the selected source files, directories and sub-directories into the project.
- **Import into Project**—Copies all of the selected source files and directories into the project directory.
- **Create Source**—Invokes the New Source File dialog box, which enables you to enter the file name, location, library and type. The new source file will be opened in the RTL Editor.
- **Set Library**—Enables you to select a library for the selected RTL source file(s).

- **Run Elaboration**—Invokes the RTL parsing and elaboration capabilities, which enable resource estimation and RTL schematic exploration.
- **Run Synthesis**—Invokes the Run Synthesis dialog box to create and launch a Synthesis Run.

Viewing Source File Properties

Selecting an RTL source file in the Sources view displays information in the Source File Properties view. The location, library, size, modified timestamp date, and parent are displayed for the file.

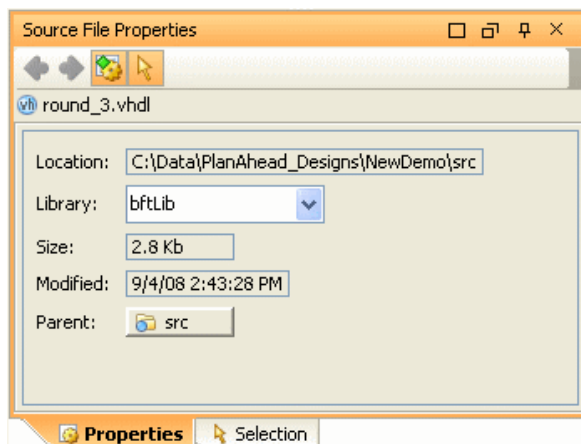


Figure 6-3: Viewing Source File Properties

A new target Library is set when you select a new one from the chooser menu, and click **Apply**.

When an RTL source displays in red, then PlanAhead could not find the required files.

Adding Sources to the Project

Adding Source Files or Directories

RTL sources can be added to the project at any time.

1. Select the **Add Sources** command to invoke the Add Sources dialog box.

In this dialog box, the buttons are defined as follows:

- ♦ **Add Files**—Invokes file browser to select RTL files to add to the project. VHDL libraries can be specified or selected at the time of import.
- ♦ **Add Directories**—Invokes directory browser to add all RTL source files from the selected directories and their sub-directories to the Project. All files in the directory hierarchy with recognized source file extensions are added to the project.
- ♦ **Remove**—Removes the selected source files from this dialog box.
- ♦ **Import Source to the Project**—Copies the original source files into the PlanAhead project and references them locally.

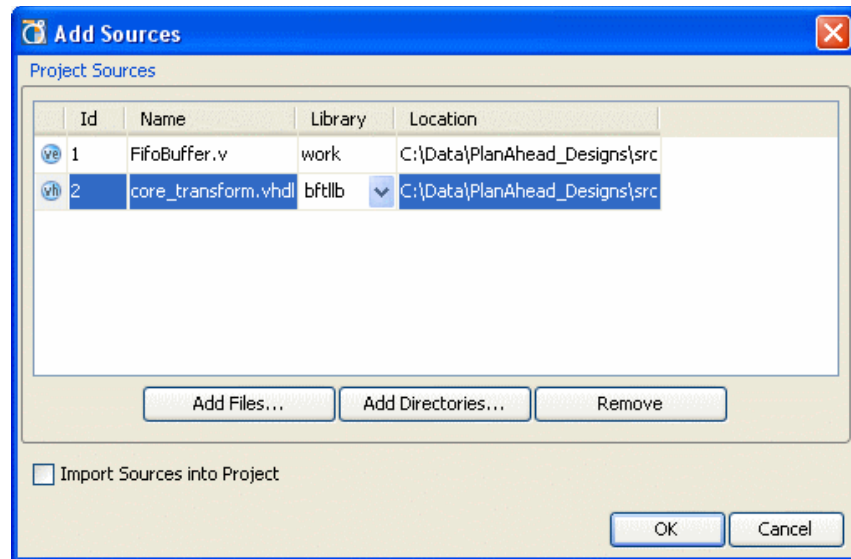


Figure 6-4: Adding Source Files and Directories to the Project

When adding sources, you can click on the Library field, and enter a library name for the file.

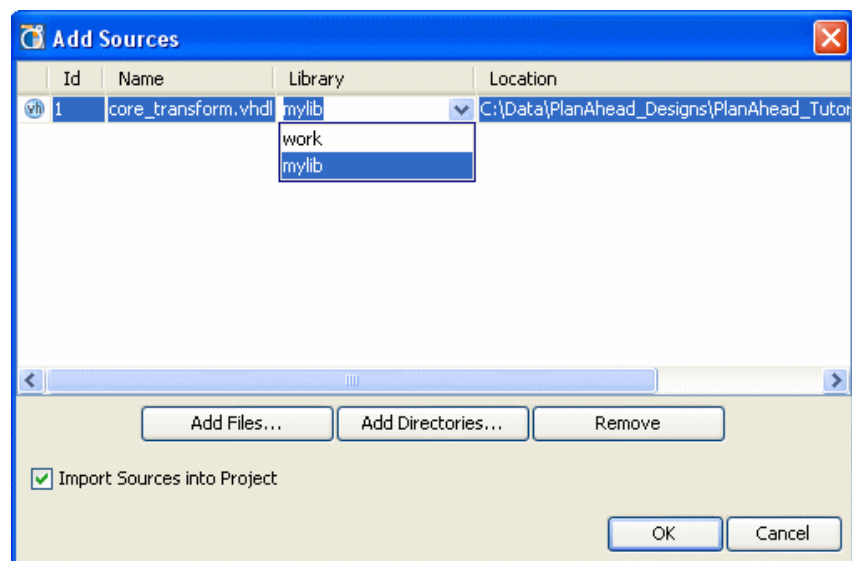


Figure 6-5: Changing Source Library in Add Sources Dialog Box

Copying Sources into Project Directory

In order to provide flexibility on how the project is managed, PlanAhead can reference either the original source files in a remote location or the source file copied to the project directory. Copying the files to the project is recommended if you plan to move or archive the PlanAhead project as all of the files are self-contained within the project. If sources are left in a remote location rather than copied, updates and maintenance can be easier as only one copy is maintained.

Sources are copied locally when you select the **Import Sources into Project** in the Add Sources dialog box.

Source files and directories can also be selected in the Sources view, and copied into the Project using the **Import into Project** popup menu command.

Creating a New Source File

New Verilog or VHDL source files can be created by using the **Create Source** command. The New Source File dialog box appears.

1. Define the following information:
 - ♦ **Name**—Enter a name for the new HDL source file.
 - ♦ **Location**—Designate a location to create the file.
 - ♦ **Type**—Select either Verilog or VHDL format.
 - ♦ **Library**—Select which library to reference for the file.
 - ♦ **File to Create**—Displays the file location and name.

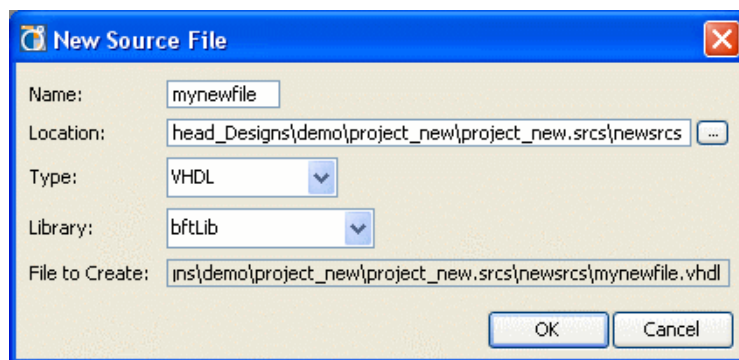


Figure 6-6: Create New Source File Dialog

2. Once the information is entered, click **OK**.

The RTL Editor opens with the newly created file. Standard HDL templates are available in the RTL Editor to assist with defining certain logic constructs. For more information, see [“Using the RTL Editor.”](#)

Updating RTL Source Files

Externally Referenced Source Files

If RTL source files are not copied to the project, the original versions are referenced by PlanAhead. Therefore, any changes made to the RTL source files are immediately recognized when the design is elaborated or when the RTL files are accessed.

Sources Copied to the Project

RTL Source files that have been copied into the project during project creation or by using the **Import to Project** command must be either maintained within PlanAhead or updated if the original versions are modified.

Updating source files can be accomplished in several ways:

- Make all RTL changes inside of PlanAhead using the RTL Editor.
or
- Select the **Remove from Project** popup menu command to delete the modified RTL sources from the project, and then select **Add Sources** to import the newly updated versions.
or
- Add the newly updated sources to the project by selecting **Add Sources**, and then select the **Disable** popup menu command to disable the outdated sources.

Using the RTL Editor

The PlanAhead RTL environment provides a robust RTL Editor for creating or modifying RTL sources. The RTL editor utilizes color coding to distinguish the various types of RTL constructs. Multiple files can be opened simultaneously. Each open file displays a view tab in the PlanAhead Workspace view, which allows easy access to all open files.

The Project environment enables cross probing to and from the RTL Editor with other views, such as the Schematic, Elaborate Results, Hierarchy and RTL Netlist views.

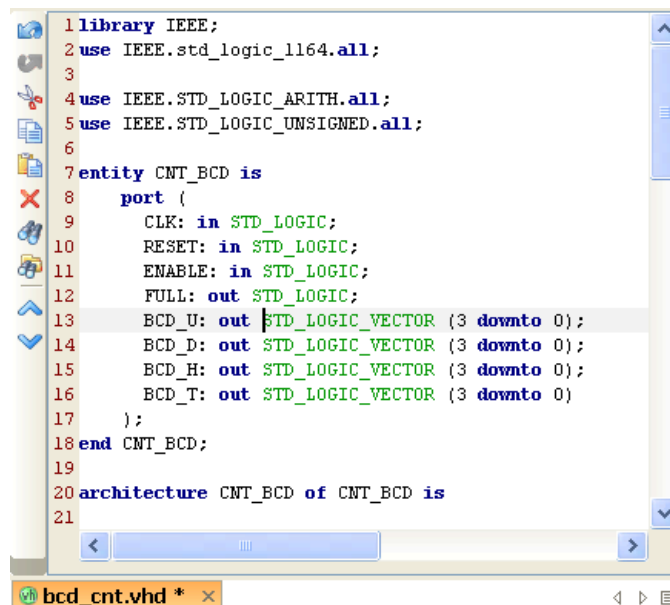


Figure 6-7: RTL Editor

An asterisk (*) is appended to the file name in the view tab if the file has been modified without saving. The file can be saved by using the **Save Project** command.

Using the RTL Editor Specific Popup Menu Commands

The RTL Editor commands and a brief description of each is as follows:

- **Cut, Copy, Paste**—Cuts or copies the selected source code into the clipboard. Pastes the contents of the clipboard.
- **Duplicate Selection**—Copies the current selected text and inserts it at the current cursor location.
- **Insert Template**—Invokes the Template browser to select a logic construct to insert in the RTL file.
- **Find, Replace**—Invokes the Find Text dialog box to enter the text string in order to filter criteria for the search. Selected text can be used to seed the search string. The Find dialog box enables you to search forward or in reverse for each occurrence, and a “replace all” option is available.
- **Find Next, Find Previous**—Performs a search based on criteria in the Find dialog box.
- **Find in Files**—Invokes the Find in Files dialog box to enter text strings to search in the selected files. The Find in Files Results view is displayed with the results of the search.
- **Indent Selection, Unindent Selection, Comment with Line Comment, Comment with Block Comment**—These self-explanatory commands perform the function described in their titles.
- **Run Elaboration**—Invokes a dialog box to enter the name of the top module and to then launch the RTL analysis routine to compile the design. Refer to [“Elaborating and Analyzing the RTL Design.”](#)

The common popup menu commands are described in the [“Using Common Popup Menu Commands,”](#) page 251.

Using the *Find in Files* Command to Search Source Files

The **Find** or **Find in Files** popup menu commands can be used to search for any given text string in a selected set of source files.

Any text string, including wildcards ‘*’, can be entered as search criteria. Filtering options are provided to limit the search based on all files in the project or all open files. The forward or backward direction of the search can also be specified.

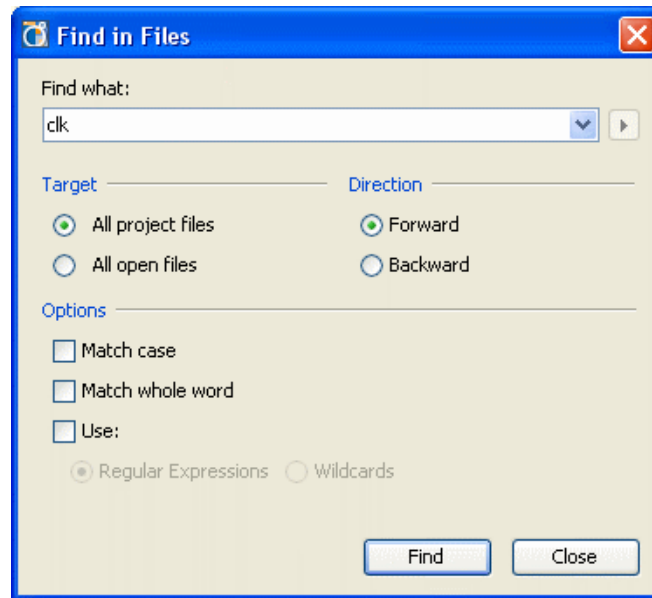


Figure 6-8: Find in Files Dialog Box

The search results are displayed in the Find in Files Results view. A list of files that contain the search string and the number of occurrences in each file are displayed. Select any occurrence in the list to load that file into the RTL Editor and highlight the string.

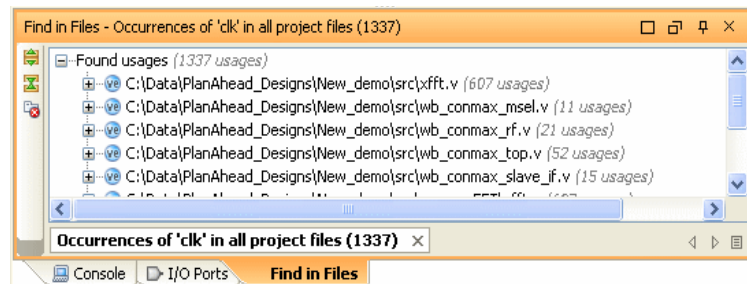


Figure 6-9: Find in Files Results View

Elaborating and Analyzing the RTL Design

PlanAhead enables you to quickly compile and analyze the RTL sources. Elaboration is available to analyze the compiled RTL design prior to running synthesis, but is not a required step prior to synthesis. All of the RTL source files imported into the project will be elaborated regardless of whether they are compiled as a part of the design during synthesis. Elaboration results are not saved with the design. Elaboration is run and rerun until the design is synthesized and a floorplan is created.

1. Once the design source files have been imported into the Project, select one of the following commands to elaborate the design.
 - ♦ Select **Tools > Run Elaboration**.
 - ♦ Select the **Run Elaboration** popup menu command.

The Run Elaboration dialog box opens.

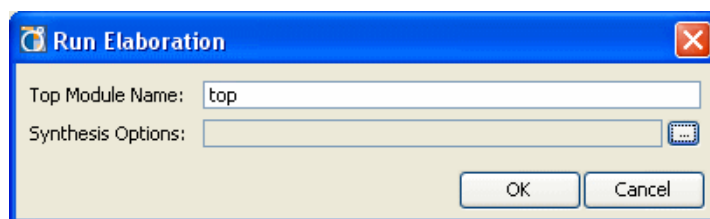


Figure 6-10: Elaborate Design Dialog Box

2. Enter the top-level module to elaborate in the Run Elaboration dialog box.
3. Optional. Click the file browser to enter synthesis options.
4. In the Synthesis Options dialog box, specify the Verilog or VHDL language options and a Loop count.

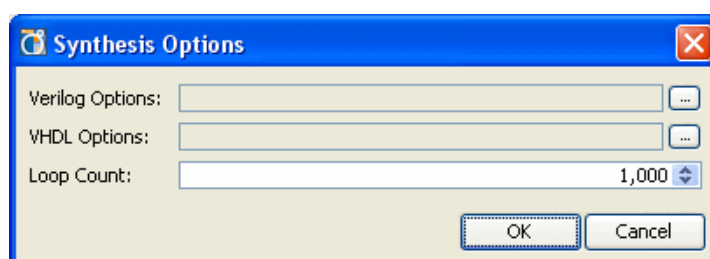


Figure 6-11: Elaboration Synthesis Options

5. Click **OK**.

Viewing Elaboration Results

The Elaboration view displays the results of the compilation flagging irregularities in the RTL Source files.

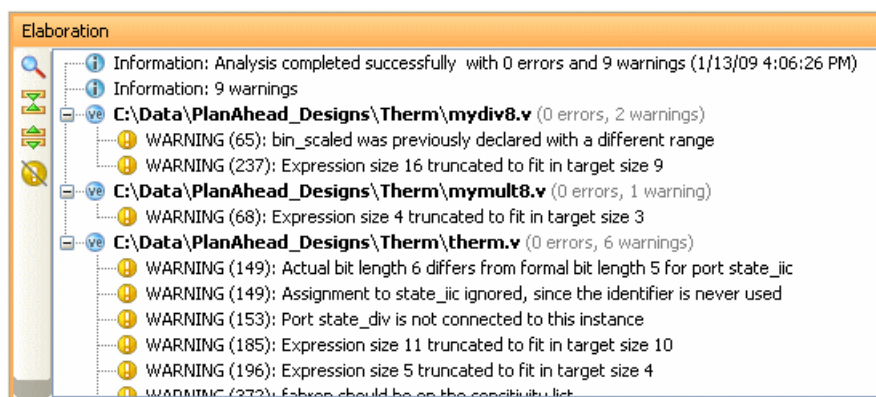


Figure 6-12: Viewing Elaboration Results

Selecting any of the Warning or Error lines in the Elaboration view will load the appropriate RTL Source file into the RTL Editor and highlight the source code in question.

Filtering the Results for Errors Only

The RTL Elaboration results can be filtered to display only error messages.

Select the Hide Warning Messages button to filter the results for errors only.



Figure 6-13: Elaboration Results: Hide Warning Messages Button

Using the RTL Netlist View

After successful elaboration, the RTL Netlist view is invoked to display the RTL logic hierarchy.

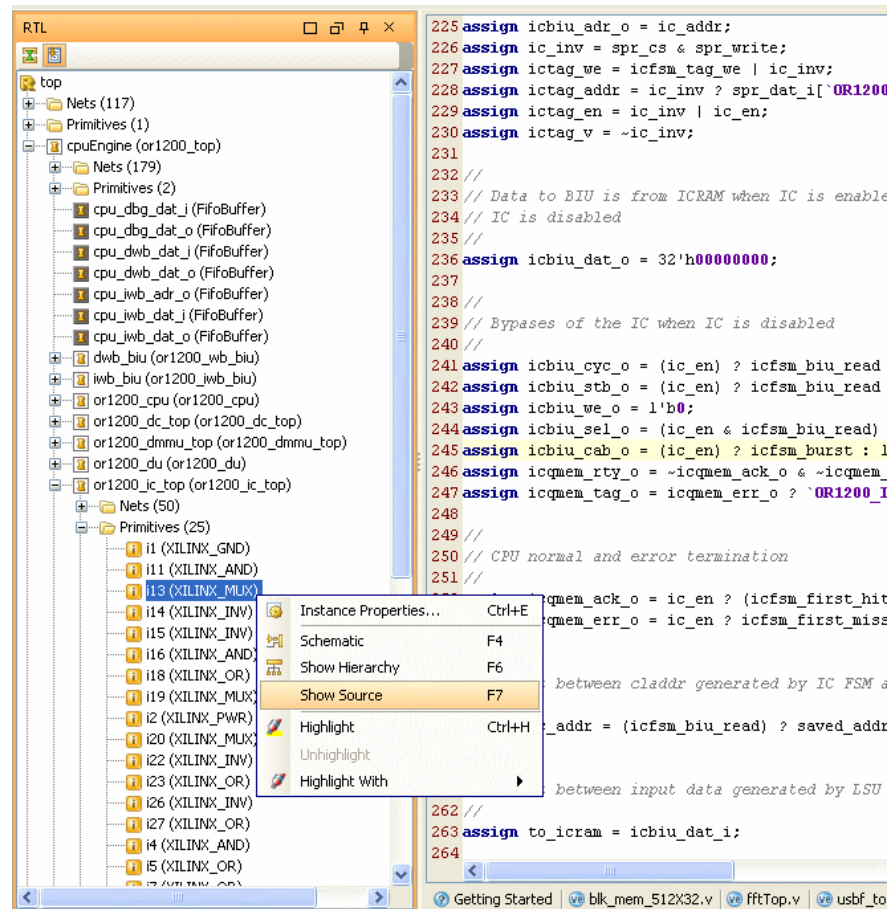


Figure 6-14: RTL View

The RTL netlist can be expanded and collapsed using the tree widget controls and view specific toolbar icons.

Using the RTL View Specific Popup Menu Commands

Common commands are described in the [“Using Common Popup Menu Commands,”](#) page 251. The RTL view commands and a brief description of each are as follows:

- **Schematic**—Invokes the RTL Schematic view.
- **Show Hierarchy**—Invokes the RTL Hierarchy view.
- **Show in Source**—Invokes the HDL Editor and highlights the selected logic.

Using the RTL Hierarchy View

The RTL netlist hierarchy can be examined using the RTL Hierarchy view. Select the **Show Hierarchy** popup menu command to invoke the RTL Hierarchy view.

The design logic hierarchy is displayed in the RTL Hierarchy view with relative sized modules.

The selected module(s) are cross highlighted in the RTL Hierarchy view.

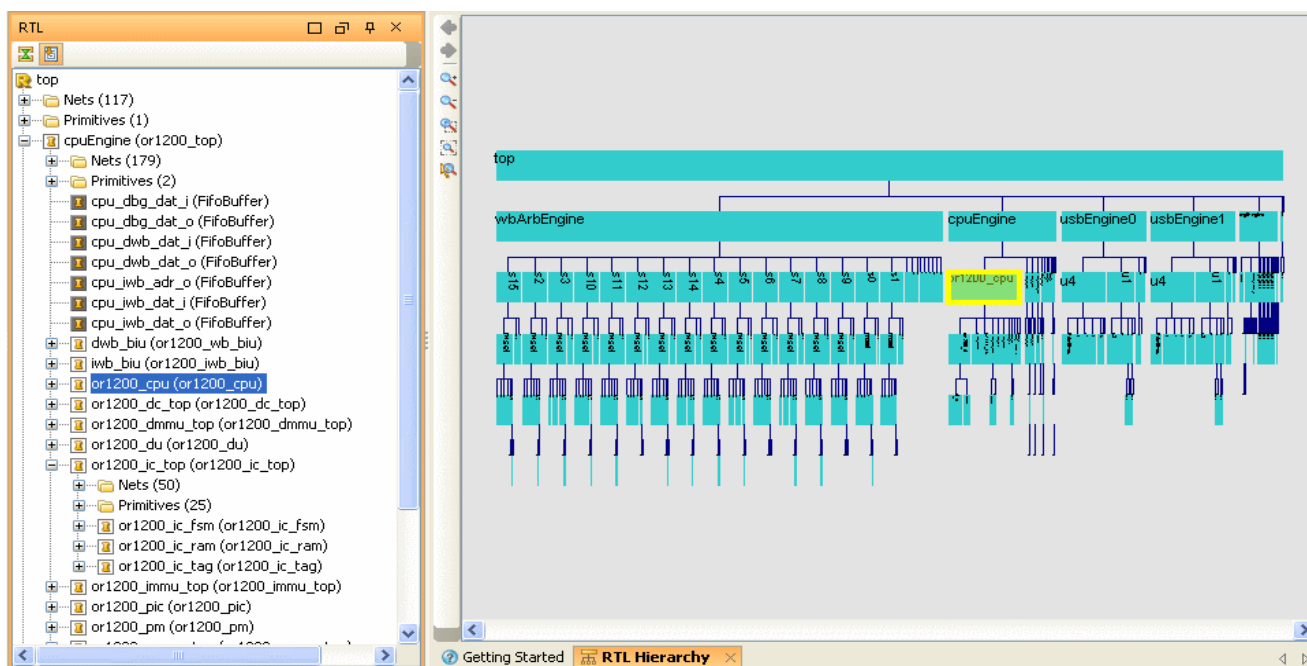


Figure 6-15: RTL Hierarchy View

The RTL Hierarchy view works similarly to the Instance Hierarchy view. For more information on exploring the RTL Hierarchy, refer to [“Using the Instance Hierarchy View,”](#) page 236.

Viewing the Resource Estimates for Modules

The RTL Netlist view works similarly to the Netlist view. Any level of logic hierarchy can be selected and its hardware resources analyzed. The Statistics tab in the Instance Properties view can be selected to view the estimated device resource requirements for the selected module or for the top-level module in the RTL view. Logic resources are categorized as Arithmetic, Comparators, Multiplexers, Storage, etc. Memory and primitive

tables are available, and they list all memories, and their depth, bit width, number of ports, etc., and macros/primitives broken down by bit width in the chosen level of the hierarchy.

The Resource Estimator provides information about hardware resources for an RTL design without running Synthesis, and therefore, with a much quicker run-time. The current accuracy is an average of +/-15%.

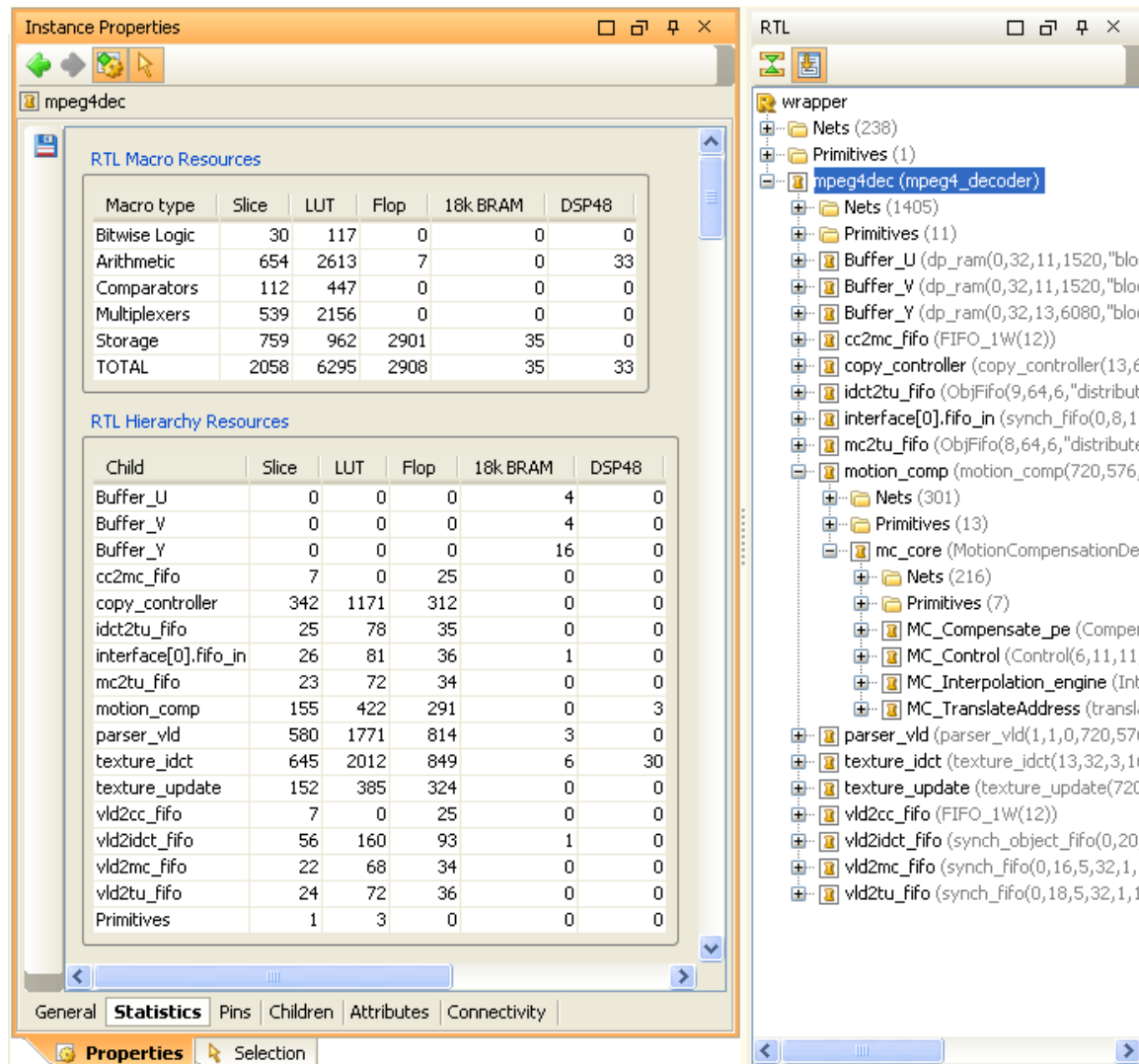


Figure 6-16: Viewing RTL Resource Estimates

Click the Save icon to save the Statistics report to XML format (for parsing) or XLS format.

Analyzing the RTL Schematic

The RTL view works similarly as the Netlist view. Any level of logic hierarchy can be selected and viewed in the RTL Schematic view. To invoke the RTL Schematic view for any selected logic, select one of the following commands:

- Click the **Schematic** toolbar button.
- Select **Tools > Schematic**.

- Select **Schematic** from the popup menu.

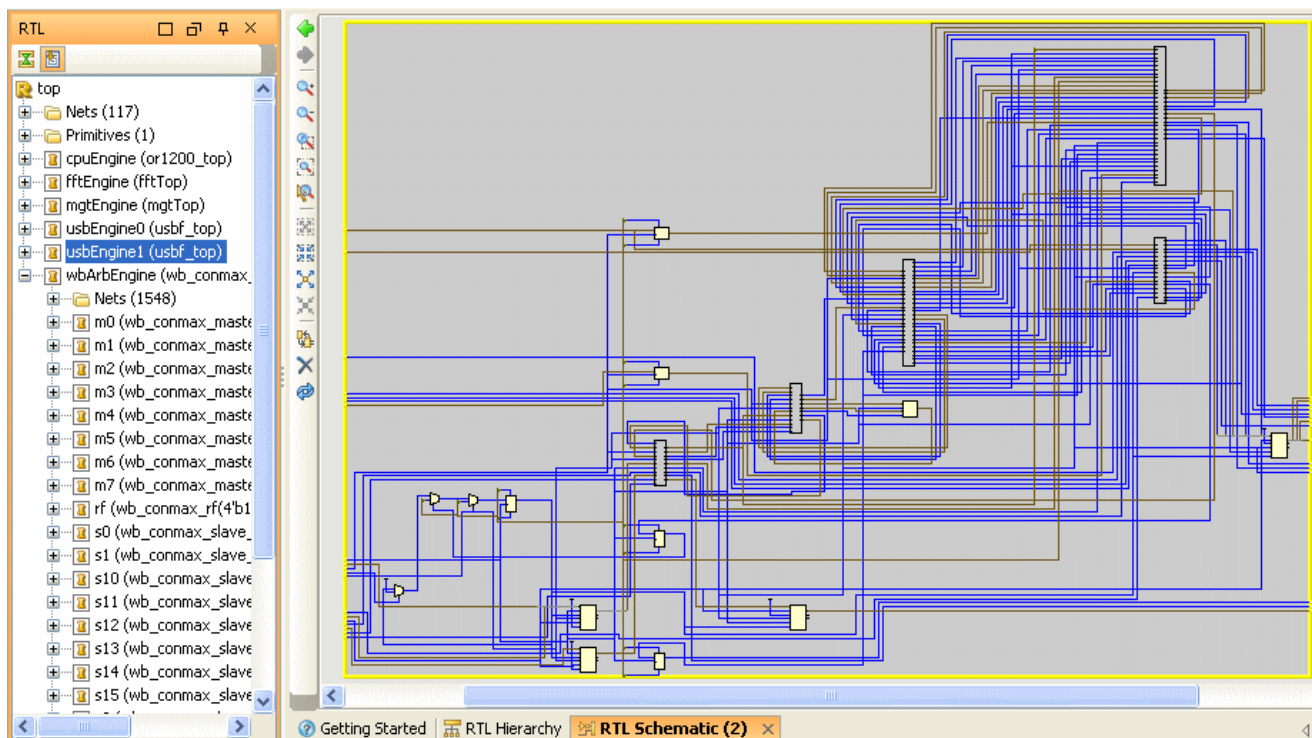


Figure 6-17: RTL Schematic View

The RTL Schematic view works similarly to the Schematic view. For more information on traversing and exploring the RTL Schematic, refer to [“Using the Schematic View,”](#) page 227.

Searching for Objects Using the Find Command

Once the RTL design has been elaborated, the Find command can be used to search for a variety of logic objects using a wide range of filtering techniques. Refer to the [“Searching for Objects Using the Find Command,”](#) page 270.

Running RTL DRCs

Selecting the DRC Rules to Run

PlanAhead offers several Design Rule Checks (DRCs) that can be run after RTL design elaboration. The available rules focus on power reduction and performance improvement opportunities.

1. Select one of the following commands to run the DRC checks after elaborating the design:
 - ◆ **Tools > Run DRC**
 - ◆ Select the **Run Design Rule Checker** toolbar button



Figure 6-18: Run Design Rule Checker Toolbar Button

2. In the Run DRC dialog box, select the desired rules.
For rule descriptions, see “Definitions of RTL DRC Rules.”
3. Click **OK**.

Analyzing DRC Violations

If violations are found, the DRC Results view will appear.

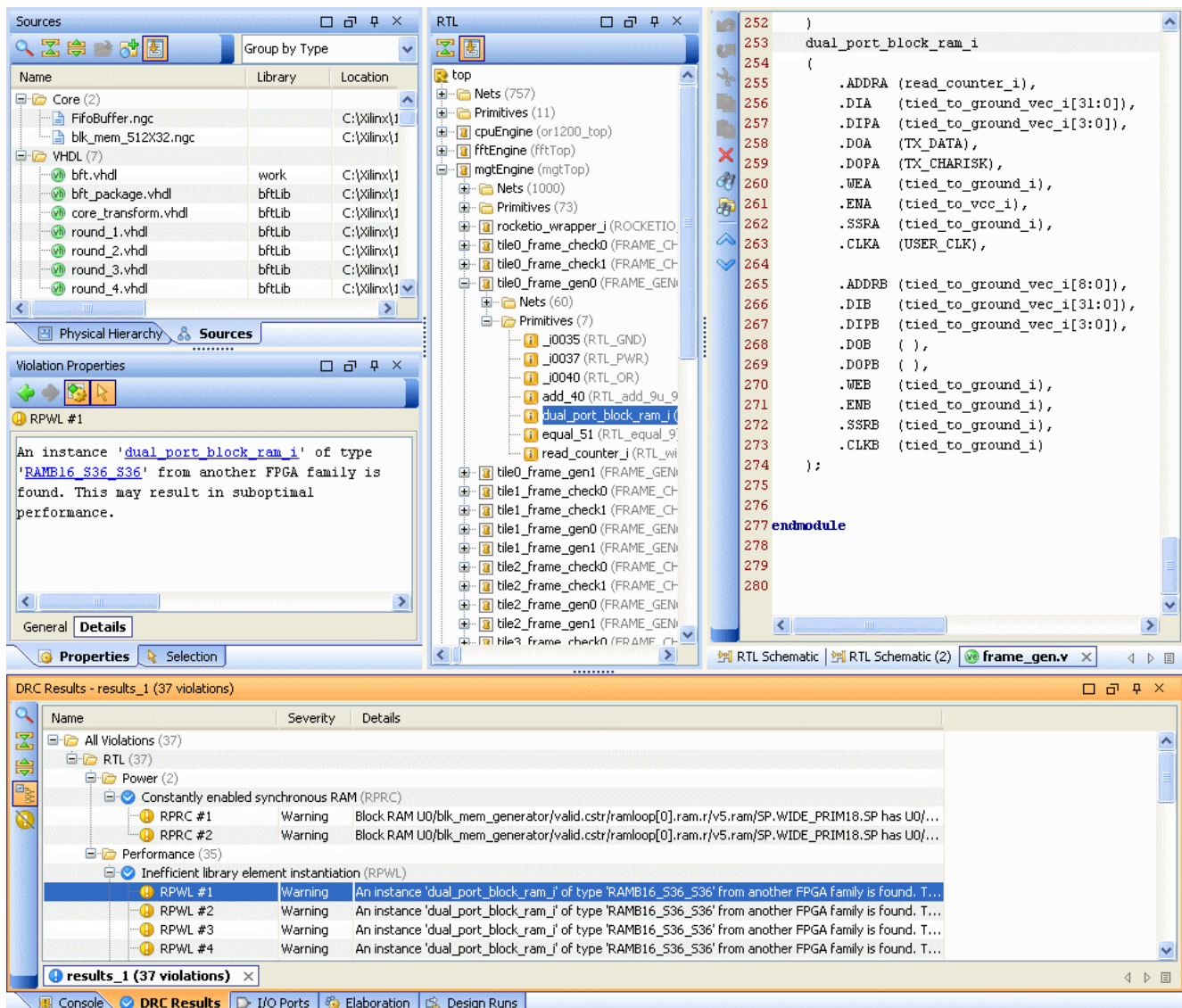


Figure 6-19: Selecting the Objects that are Violating DRCs

Some of the navigation functions in this view are as follows:

- Select any of the violations to display information about the violation in the Violations Properties view.

- Select the links in the Violations Properties view to highlight the design objects in question.
- Select the **Show Source** popup menu command to highlight the line of RTL source.

Click the Hide Warning and Information Messages toolbar button to hide all warnings and info messages and view only errors. Click the toolbar button again to view all errors and warnings once again.



Figure 6-20: Hide Warning and Information Messages Button

Definitions of RTL DRC Rules

The following tables describe the various DRC rules, rule intent and severity.

RTL Rules

Table 6-1: Power Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Constantly enabled synchronous RAM	RPRC	A described RAM (either inferred or instantiated), which are constantly enabled, was found in one or both ports. If it can be determined that this RAM is not constantly accessed. Significant power reduction may be seen by describing the logic to disable the RAM unless it is being accessed.	Warning
Inefficient dangling BRAM port	RPRM	A RAM in which there is an unconnected output port has been detected, and the WRITE_MODE is set to a value other than NO_CHANGE. Modifying the description of the RAM in order to set unconnected output port (WRITE_MODE set to NO_CHANGE) may save up to 10% of the dissipated BlockRAM power.	Warning
Shallow RAM implemented in Block RAM	RPRS	Virtex®-5 and Virtex-6 devices: For wide (over 18-bits) and shallow (64-bits or less) RAM, it is generally advantageous to choose SelectRAM (LUT-based RAM referred to as distributed RAM) whenever possible unless the RAM is being used as a FIFO in which case the cross over point becomes a depth of 32-bits or less. When building interfaces less than 18-bits wide, the LUT-based SelectRAM could be a better choice for depths up to 128-bits; however, generally past that, the dedicated BlockRAM is a better choice for power.	Warning
Inefficient mapping of small multiplier in DSP block	RPDS	Small multipliers mapped to DSP or to other hard multipliers IP, such as MULT18X18, should be pushed to MSBs. The rest of the LSBs should be mapped to ground. In this way, the carry propagation is reduced to its minimum. Usual implementation, especially when inferring the multiplier, uses LSBs and sign extensions to map the MSBs.	Warning

Table 6-2: Performance Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Inefficient library element instantiation	RPWL	Found instance "<instance name>" of type "library_component_name" which belongs to another FPGA family. This may result in suboptimal performance. The ISE® software may automatically remap this element onto a similar element in the selected family. However, modifying the source code to infer or instantiate native elements will take advantage of any added or expanded functionalities in the element. This may in turn improve area utilization, performance, etc.	Warning
Missing pipeline register	RPPR	Found multiplier with unregistered outputs. You can improve the multiplier clock to out performance by adding a level of registers. In addition, for best results, avoid using asynchronous control signals on these registers. Found RAM/ROM with unregistered outputs. You can improve the RAM/ROM clock to out performance by adding a level of registers. In addition, for best results, avoid using asynchronous control signals on these registers.	Warning
In efficient pipeline register	RPIP	Found <register_name> (<file_name>:<line_number>) register with asynchronous control signals on input or output of multiply function. Dedicated DSP hardware resources do not have asynchronous control signals, such as preset or clear. The registers will not be mapped into the dedicated hardware resources resulting in suboptimal use of the device.	Warning
Found Black Box instance not belonging to UNISIM library	RPBX	Component/Module <component/module_name> description unavailable during synthesis (<file_name:line>). Paths to and from this back box cannot be optimized. Synthesis tool utilization estimates and mapping decisions may be negatively affected.	Warning

Table 6-2: Performance Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Found latch in design	RPLD	Found latch description for signal <signal_name> (<file_name>:<line_num>). Latches creates difficult to analyze timing paths which require post implementation simulations to ensure implemented design match expected behavior.	Warning
Found combinatorial loop in design	RPCL	Found combinatorial loop for signal <signal_name> (<file_name>:<line_number>). Combinatorial loops are generated when a cone of combinatorial logic uses its outputs to feed back as partial input to the same cone of logic. The total combinatorial delay from source to destination should be increased by the feedback path delay. This type of structure may be required from the design expected behavior or may be unintentional.	Warning

Implementing a Design

The PlanAhead™ software includes a synthesis and implementation environment that allows multiple synthesis and implementation attempts using different software command options, and timing or physical constraints. The synthesis and implementation attempts or “Runs” can be queued to launch sequentially or simultaneously with multi-processor machines. Synthesis runs use the Xilinx® XST synthesis tool.

You can create and save “Strategies”, which are a set of option configurations for each implementation command. These Strategies are then applied to Runs for synthesis or implementation using ISE® tools.

You can monitor progress, view log reports, and quickly identify and import the best synthesis and implementation results.

This chapter contains the following sections:

- [“Running Synthesis”](#)
- [“Running Implementation”](#)
- [“Monitoring and Configuring Runs”](#)
- [“Managing Runs”](#)
- [“Importing Run Results”](#)
- [“Running Bitgen on an Implementation Run”](#)
- [“Creating Strategies”](#)
- [“Executing Runs on Multiple Linux Hosts”](#)
- [“Interfacing with ISE Outside of PlanAhead”](#)

For more information about saved Strategies, see [“Outputs for Environment Defaults.”](#) For more information about the exported files, see [“Outputs for ISE Implementation.”](#)

Running Synthesis

Creating and Launching Synthesis Runs

Synthesis Runs can be created and launched simultaneously, or created, configured and launched independently.

Creating and Launching a Single Synthesis Run

1. Select one of the following commands:
 - ♦ Select the **Run Synthesis** main toolbar button.



Figure 7-1: **Run Synthesis Main Toolbar Button**

- ♦ Select the **Tools > Run Synthesis** command.

The Run Synthesis dialog box appears.

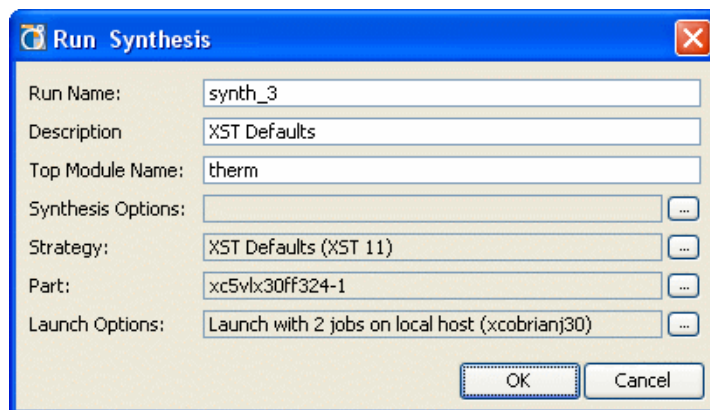


Figure 7-2: **Run Synthesis dialog box**

The Run Synthesis dialog box contains several options and editable fields:

- **Run Name**—Enter a name for the synthesis run or accept the default name.
- **Description**—Enter any description for the synthesis run.
- **Top Module Name**—Enter or accept the top-level RTL module name for the design.
- **Synthesis Options**—Click the file browser, and in the Synthesis Options dialog box, enter an optional top-level VHDL Library name, Verilog or VHDL Options or a Loop Count.
- **Strategy**—Select the Synthesis strategy to use for the Run. You can select an XST version 10 or 11 strategy. For more information, see [“Creating Strategies.”](#)

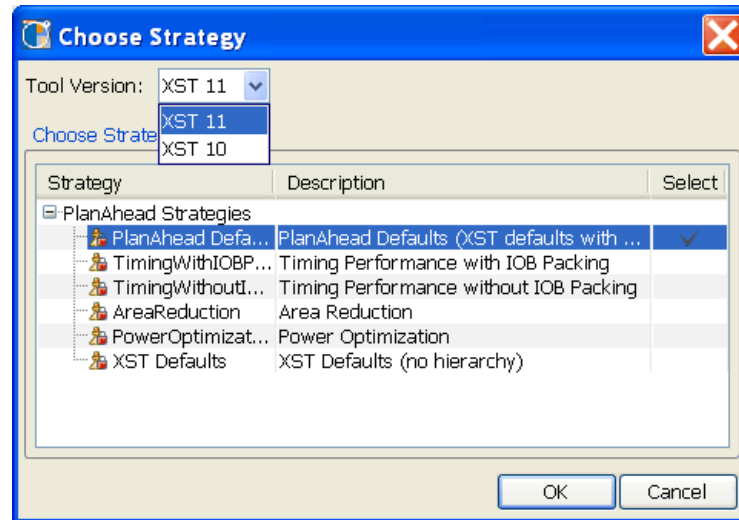


Figure 7-3: Choose a Synthesis Strategy

- **Part**—Select a target part or accept the default.
- **Launch Options**—Select additional launch options.

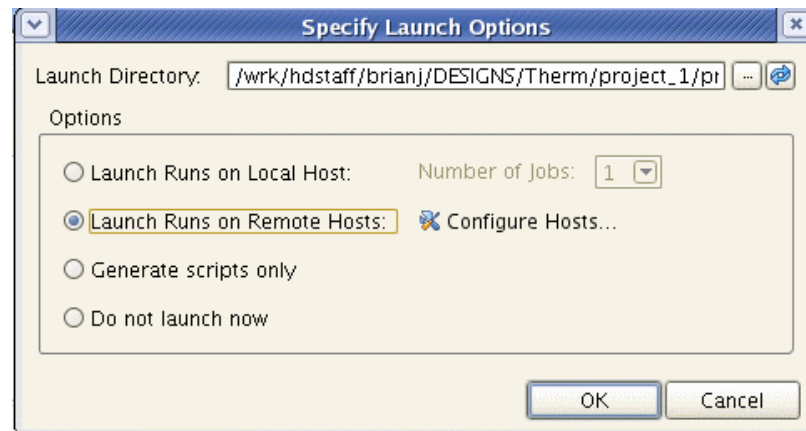


Figure 7-4: Synthesis Run Launch Options

The Specify Launch Options dialog box provides the following launch options:

- ♦ **Launch Directory**—Specify a location to create and store the synthesis run data.
Note: Defining any non-default location outside of the project directory structure makes the Project non-portable as absolute paths are written into the project files.
- ♦ **Launch Runs on Local Host**—Select this option to launch the Run on the local machine processor.
 - **Number of Jobs**—Define the number of local processors to use for Runs. This option is only used when launching multiple runs simultaneously. Individual runs will be launched on each processor. No multi-threaded processors are used with this option.
- ♦ **Launch Runs on Remote Hosts** (Linux only)—Select this option to use remote hosts to launch job or jobs.

- **Configure Hosts**—Select this option to configure remote hosts. Refer to [“Executing Runs on Multiple Linux Hosts.”](#)
 - ◆ **Generate scripts only**—Select this option to export and create the run directory and run script, but not to launch the run from PlanAhead. The script can be run at a later time outside of the PlanAhead environment.
 - ◆ **Do not launch the run**—Select this option to Create the Run in the Design Runs view, but not to export the data or launch the Run yet.
2. Click **OK** to create the Run with the selected launch options.

Once the Run is complete, the following dialog box is displayed prompting you to take the next step.

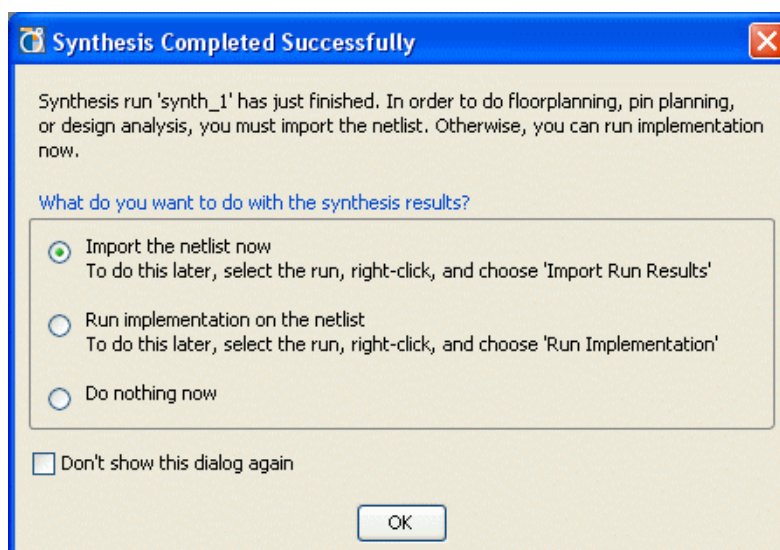


Figure 7-5: **Synthesis Completely Successfully Dialog Box**

3. In the Synthesis Completed Successfully dialog box, select the option that matches how you want to proceed.
 - ◆ **Import the Netlist Now**—Will import the netlist into the PlanAhead design analysis and floorplanning environment enabling you to perform I/O pin planning, design analysis and floorplanning.
 - ◆ **Run Implementation on the Netlist**—Will launch the Run Implementation dialog enabling you to create and launch an implementation run.
 - ◆ **Do Nothing Now**—Will enable you to continue creating and launching synthesis or implementation runs using the PlanAhead environment commands.

Creating Multiple Synthesis Runs

PlanAhead enables you to create and launch multiple synthesis runs simultaneously. Various synthesis options and tools can be explored to find the best results.

1. Select **Tools > Run Multiple Strategies** to create multiple synthesis Runs.
2. If prompted, select **Synthesis** from the first Run Multiple Strategies dialog box.

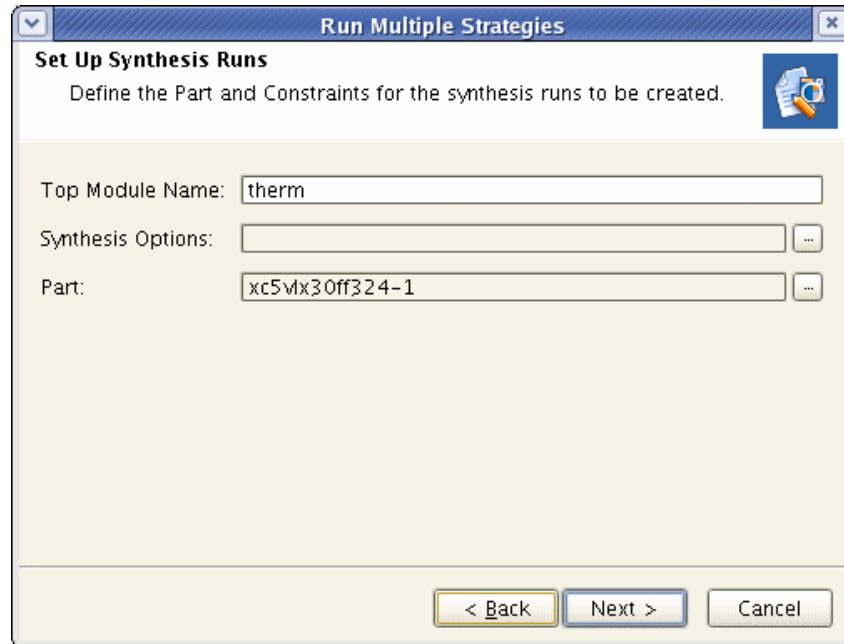


Figure 7-6: Set Up Multiple Synthesis Runs

The Set Up Synthesis Runs dialog box has several options fields.

- ◆ **Top Module Name**—Enter or accept the top-level RTL module name for the design.
 - ◆ **Synthesis Options**—Click the file browser button, and enter an optional top-level VHDL Library name, Verilog or VHDL Options or a Loop Count in the Synthesis Options dialog box.
 - ◆ **Part**—Select a target part or accept the default.
3. Click **Next** to bring up the Choose Strategy dialog box.

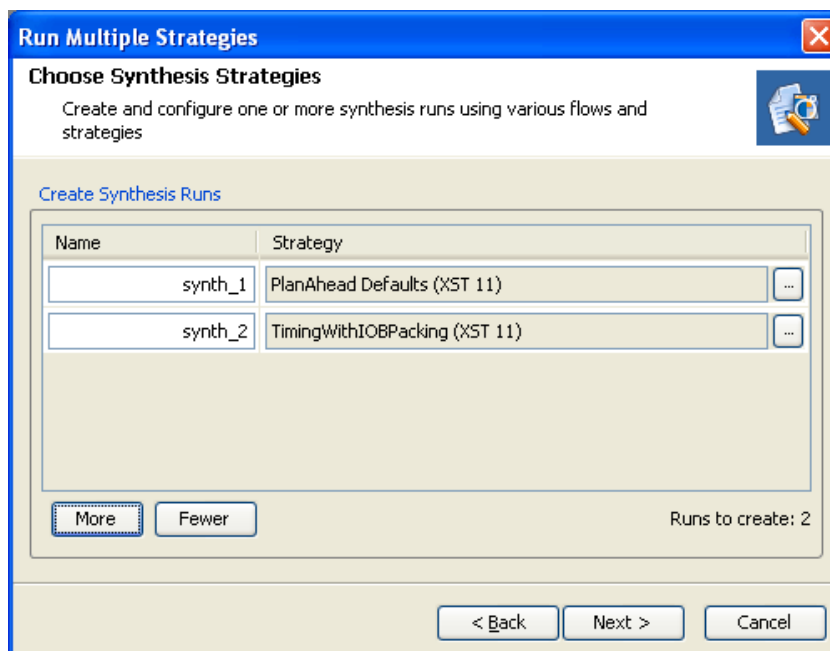


Figure 7-7: Choose Synthesis Strategy

4. Select a Name and Strategy for the first synthesis Run.
5. Select the **More** button to add more runs.
6. Enter names and choose a synthesis strategies for the additional Runs.
7. Click **Next** to invoke the Launch Options dialog box. For more information on specifying Launch Options, see [“Creating and Launching a Single Synthesis Run.”](#)
8. Click **OK** to create the defined Runs and execute the specified Launch options.

Synthesis Methodology Tips

The following are suggestions on a logic synthesis methodology. For more information on optimizing ISE synthesis results, see the Xilinx *Synthesis and Simulation Design Guide*.

1. To the extent possible, partition the design at the RTL level such that critical timing paths are confined to individual modules. Critical paths that span large numbers of hierarchical modules can be difficult to floorplan.
2. Register the outputs of all the modules to help limit the number of modules involved in a critical path.
3. Long paths in single large hierarchical blocks can make floorplanning a difficult task. Consider dividing large hierarchical blocks in the RTL.
4. If the design is expected to change often, consider an incremental approach to synthesis. In an incremental approach, individual blocks can be synthesized separately or the synthesis attributes (SYN_HIER=HARD) can be used to preserve the hierarchy. Hierarchy preservation will help an incremental flow but may hurt performance since global optimizations across hierarchy are disabled. This trade-off needs to be considered before you embark on an incremental RTL synthesis methodology.
5. Constrain the synthesis engine to rebuild or to otherwise preserve the hierarchy in the synthesized netlist. Flattened netlists may be optimal from a synthesis perspective, but they make it very difficult to reliably floorplan and constrain placement. Consider

using the options and synthesis pragmas to rebuild the hierarchy, for example, the XST command line option `-netlist_hierarchy = rebuilt`, which is available in 9.2 and later versions.

Running Implementation

PlanAhead refers to the process of running the ISE command sequence of *ngdbuild*, *map*, *par*, *trce* and *xdl* as “implementation”. The *bitgen* command can be run separately from the PlanAhead environment after satisfactory implementation results are achieved. PlanAhead automatically runs these commands and enables you to review and analyze the results. The run data is all stored in the project directory for future retrieval of bit files.

Creating and Launching Implementation Runs

Implementation Runs can be created and launched simultaneously, or created, configured and launched independently.

Creating and Launching a Single Implementation Run

1. Select one of the following commands:
 - ♦ Select the **Run Implementation** main toolbar button.



Figure 7-8: Run Implementation Main Toolbar Button

- ♦ Select the **Tools > Run Implementation** command.

The Run Implementation dialog box appears.

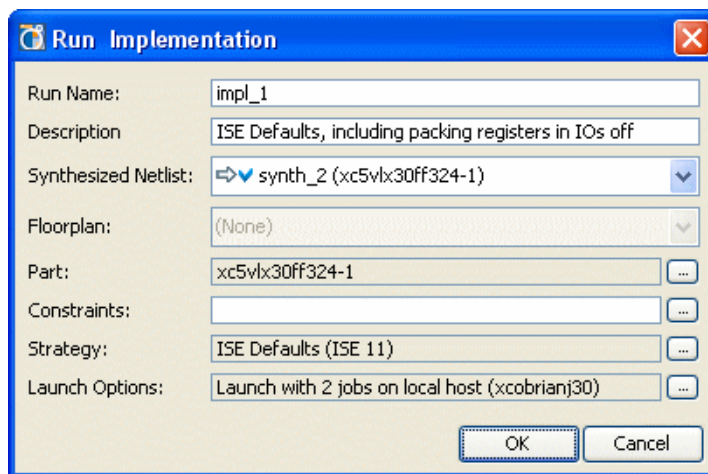


Figure 7-9: Run Implementation

The Run Implementation dialog box contains several options and editable fields:

- ♦ **Run Name**—Enter a name for the implementation run or accept the default name.
- ♦ **Description**—Enter any description for the implementation run.
- ♦ **Synthesized Netlist**—Select the name of the synthesis run to implement.
- ♦ **Floorplan**—Select the name of the Floorplan to implement.

- ◆ **Part**—Select a target part or accept the default.
- ◆ **Constraints**—Browse to select a single top-level UCF constraint file to use for the run. This field is disabled if a floorplan is selected as the constraints from the floorplan will be used.
- ◆ **Strategy**—Select the implementation strategy to use for the Run. For more information, see “Creating Strategies.”

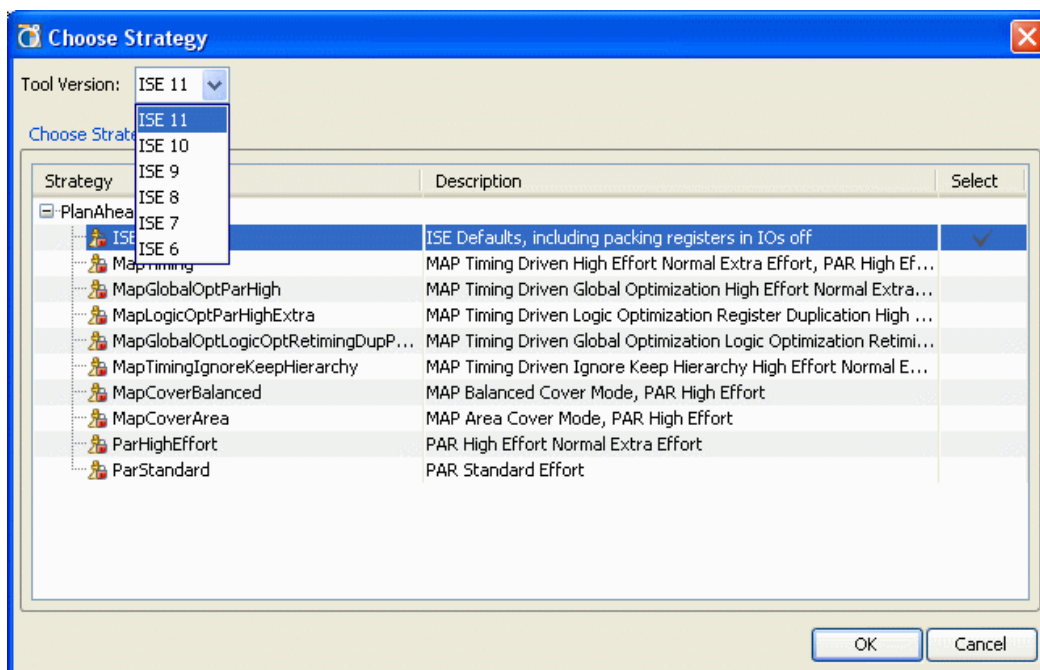


Figure 7-10: Choose an Implementation Strategy

- ◆ **Launch Options**—Select additional launch options.

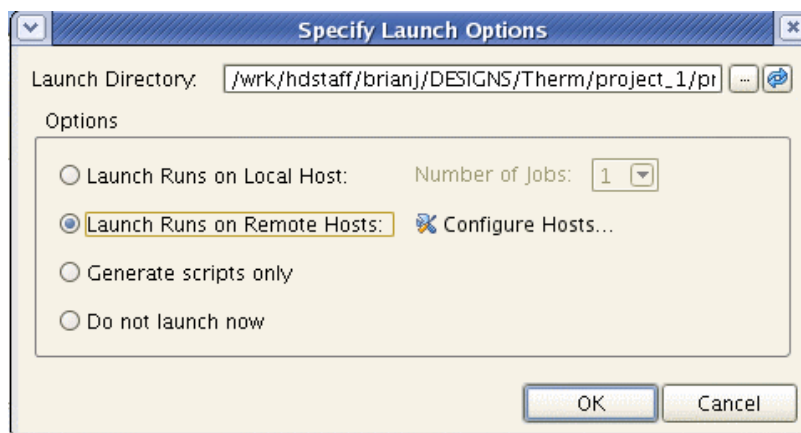


Figure 7-11: Implementation Run Launch Options

The Specify Launch Options dialog provides the following launch options:

- ◆ **Launch Directory**—Specify a location to create and store the implementation run data.

Note: Defining any non-default location outside of the project directory structure makes the project non-portable as absolute paths are written into the project files.

- ◆ **Launch Runs on Local Host**—Select this option to launch the Run on the local machine processor
 - **Number of Jobs**—Define the number of local processors to use for Runs. This option is only used when launching multiple runs simultaneously. Individual runs will be launched on each processor. No multi-threaded processors are used with this option.
 - ◆ **Launch Runs on Remote Hosts** (Linux only)—Select this option to use remote hosts to launch job or jobs.
 - **Configure Hosts**—Select this option to configure remote hosts. For more information, see [“Executing Runs on Multiple Linux Hosts.”](#)
 - ◆ **Generate scripts only**—Select this option to export and create the run directory and run script, but not to launch the run from PlanAhead. The script can be run at a later time outside of the PlanAhead environment.
 - ◆ **Do not launch the run**—Select this option to Create the Run in the Design Runs view, but not to export the data or launch the Run yet.
2. Click **OK** to create the Run with the selected launch options.

Creating Multiple Implementation Runs

PlanAhead enables you to create and launch multiple implementation runs simultaneously. Various implementation options can be explored to find the best results.

1. Select **Tools > Run Multiple Strategies** to create multiple implementation Runs
2. Select **Implementation** from the first Run Multiple Strategies dialog box.

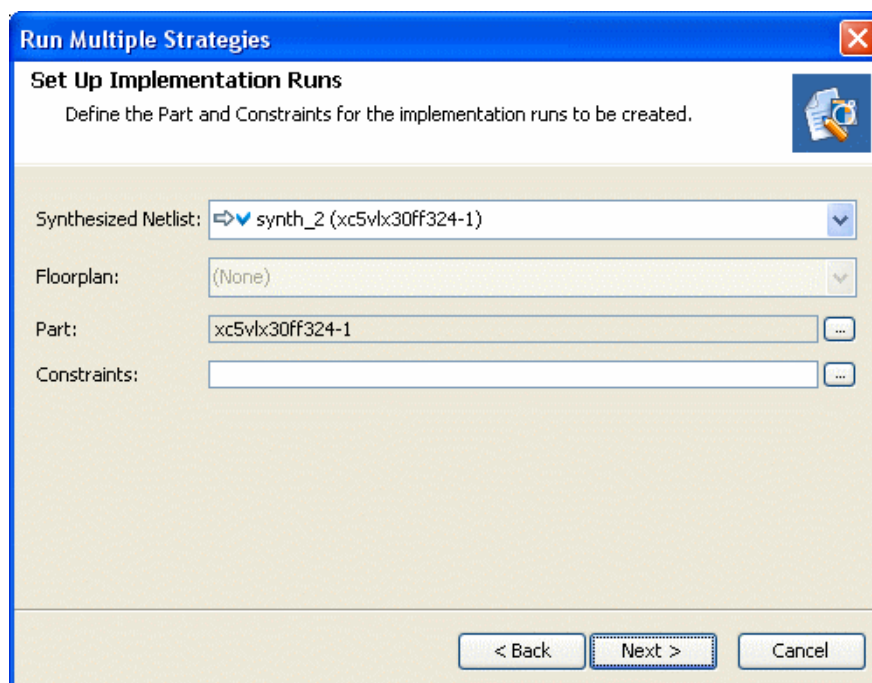


Figure 7-12: Set Up Multiple Implementation Runs

The Set Up Implementation Runs dialog box has several options fields.

- ◆ **Synthesized Netlist**—Select the name of the synthesis run to implement.
 - ◆ **Floorplan**—Select the name of the floorplan to implement.
 - ◆ **Part**—Select a target part or accept the default.
 - ◆ **Constraints**—Browse to select a single top-level UCF constraint file to use for the run. This field is disabled if a floorplan is selected as the constraints from the floorplan will be used.
3. Click **Next** to bring up the Choose Implementation Strategies dialog box.

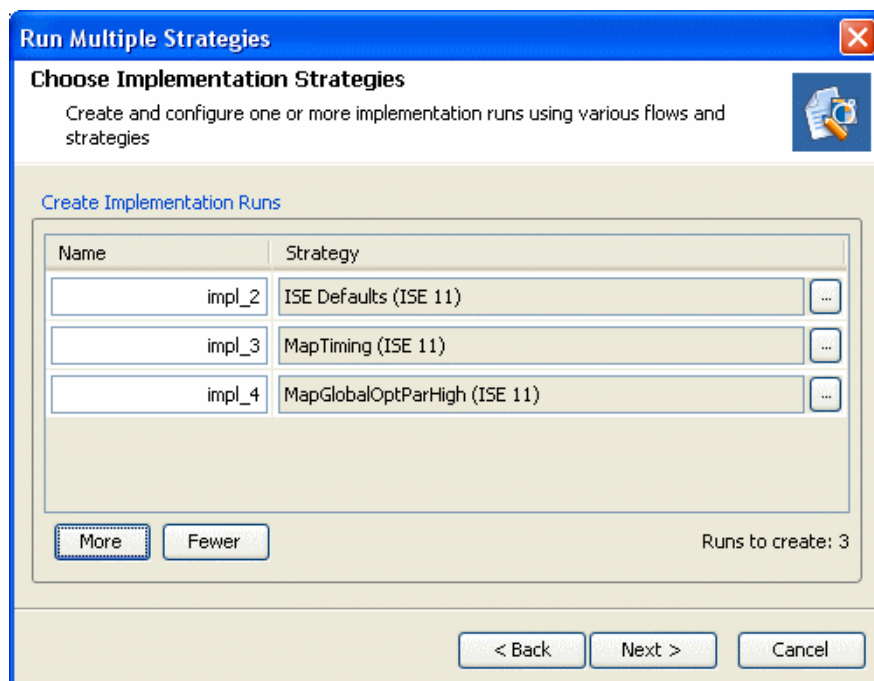


Figure 7-13: Run Multiple Strategies dialog box

4. Enter a name and select a strategy for the first implementation run.
5. Click the **More** button to add more runs.
6. Enter names and choose a implementation strategies for the additional runs.
7. Click **Next** to invoke the Launch Options dialog box. For more information on specifying Launch Options, see [“Creating and Launching a Single Implementation Run.”](#)
8. Click **OK** to create the defined Runs and execute the specified launch options.

Implementing Pblocks

PlanAhead has the unique ability to implement Pblocks individually. The ISE software **-u** option is used to isolate the module for ISE implementation. This presents many benefits when trying to achieve maximum design performance, as described below.

Implementing a standalone Pblock enables ISE to maximize the performance targets for that logic. This provides a great indicator of how the module may perform in a best-case scenario. If timing targets are not met for the standalone Pblock, timing may be extremely difficult to meet in the context of the entire design.

For logic in the design that is prone to performance-related issues, you can implement that piece of logic, lock down the implemented logic, and then implement the top-level design. With placement locked down, the chances of maintaining performance increase dramatically. There is a chance that interconnects may not follow the same routing channel tracks as expected during top-level implementation, in which case, routing delays may vary. For more information on logic locking techniques, see [“Working with Placement LOC Constraints,”](#) page 315.

Design runs can be created and configured for individual Pblocks that leverage the same monitoring options that PlanAhead provides for floorplan implementation. You can create implementation runs for one or more selected Pblocks.

Note: Before implementing Pblocks with ISE, the following Xilinx environment variable should be set to prevent logic trimming into the Pblock:

XIL_MAP_NOCLIP_ON_ALL_SIGS_U with a value of **1**

Creating Runs for Selecting Pblocks

The Run Implementation command can be seeded with pre-selected Pblocks prior to running the command.

To run the ISE commands on selected Pblocks:

1. Select a Pblock in PlanAhead.
2. Select one of the following commands:
 - ♦ Select **Tools > Run Implementation**.
 - ♦ Click the **Run Implementation** toolbar button.



Figure 7-13: Run Implementation Toolbar Button

In the Run Implementation dialog box, the Pblock field should display the selected Pblock.

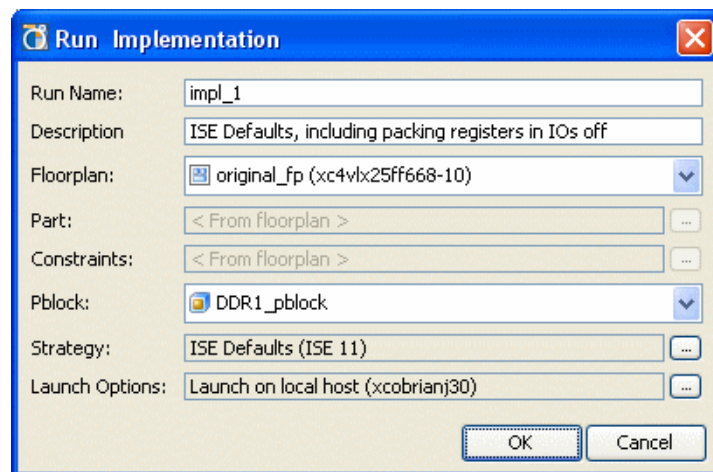


Figure 7-14: Run Implementation for Pblocks

3. Alternately, you can select the desired Pblock to implement after selecting the Run Implementation command. If a Pblock is not pre-selected, click the **Pblock** field, and select one Pblock from the list.

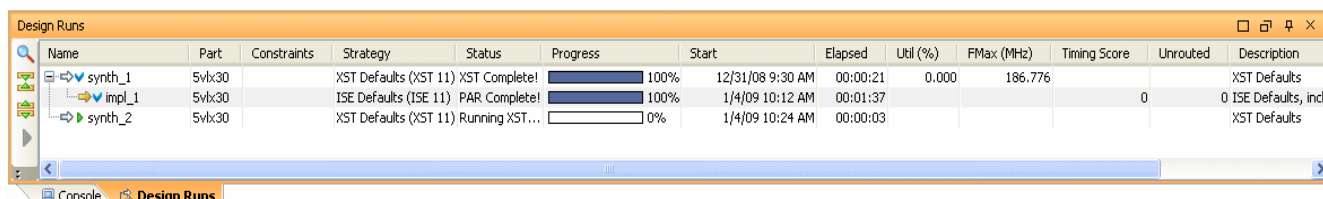
4. Select the Run options as described in “Running Implementation.”
5. Click **OK**.

The Pblock Run will appear in the Design Runs view, and can be monitored and managed as other implementation runs are.

Monitoring and Configuring Runs

Using the Design Runs View

As Runs are created, launched or imported, the status is displayed in the Design Runs view. Select **Tools > Design Runs** to invoke the Design Runs view.



Name	Part	Constraints	Strategy	Status	Progress	Start	Elapsed	Util (%)	FMax (MHz)	Timing Score	Unrouted	Description
synth_1	5vtx30		XST Defaults (XST 11)	XST Complete!	100%	12/31/08 9:30 AM	00:00:21	0.000	186.776			XST Defaults
impl_1	5vtx30		ISE Defaults (ISE 11)	PAR Complete!	100%	1/4/09 10:12 AM	00:01:37			0	0	ISE Defaults, inclu
synth_2	5vtx30		XST Defaults (XST 11)	Running XST...	0%	1/4/09 10:24 AM	00:00:03					XST Defaults

Figure 7-15: Design Runs View

The view displays the status and results of the Runs defined as well as provides commands to modify, import, launch, and manage the Runs. This same view is also used to manage and report synthesis and implementation runs.

The view indicates the runs currently running with a green arrow icon, as seen above on the left. Completed Runs receive a blue check mark icon. Run information is displayed as the commands are being run. PlanAhead can be closed without affecting runs in progress. When the Project is re-opened, the Run status is updated to reflect the latest status, which is displayed in the Design runs chart.

Columns that are used for tracking information are as follows:

- Run Name—Displays run name.
- Target Part—Indicates the target part selected for the run.
- Constraint file—Displays either the constraint file or the floorplan name used for the run.
- Strategy—Displays the strategy assigned to the Run. Strategies appearing with an “*” indicate that the command option values in the Strategy have been overridden in the Run Properties Options tab.
- Status—Indicates run status or the command that is currently running.
- Progress—Indicates overall progress of the entire ISE command sequence from *ngdbuild* through *xdl*. The Progress bar is non-linear, in that some steps may take considerably longer than others.
- Start—Indicates the time ISE started working on the design.
- Elapsed—Indicates the total elapsed time for all ISE commands run on the design.
- Device Utilization (for Synthesis runs only) —Indicates the resulting LUT utilization for the run.
- Fmax (for Synthesis runs only)—From the XST synthesis report, indicates the expected clock frequency for the run .

- **Timing Score** (for Implementation runs only)—Indicates the current Timing Score on the run in progress or after completion.
- **Unrouted Nets** (for Implementation runs only)—Indicates the current number of unrouted nets on the run in progress or after completion.
- **Description**—Displays the description associated with the run. This description is initially set to a strategy's description when that strategy is applied to the run; however, the description can be modified later.

The table is updated dynamically as the Run commands are progressing. Runs that are launched outside of PlanAhead using the PlanAhead generated scripts will cause the table to update upon invoking PlanAhead.

Using the Design Runs View Popup Menu Commands

The Design Runs view popup menu contains the following commands:

- **Synthesis or Implementation Run Properties**—Displays the Run Properties View. For more information, see [“Viewing and Modifying Run Properties.”](#)
- **Delete**—Deletes the selected Runs. You are prompted to confirm prior to removing any runs.
- **Apply Strategy**—Invokes the Strategy chooser to select a new run Strategy. This command is only available prior to launching the run.
- **Save as Strategy**—Enables you to save any modifications made to the applied strategy to a new strategy file for future use.
- **Edit Strategies**—Invokes the Tools > Options > Strategies dialog box to edit or create Strategies.
- **Launch Runs**—Invokes the Launch Runs dialog box to launch the selected runs.
- **Reset Runs**—Invokes the Reset Runs dialog box to remove previous run results and to set the Run status back to Not Started for the selected runs.
- **Import Run**—Loads the resulting netlist from the synthesis run or implementation results from ISE in to the PlanAhead analysis environment. The active loaded run appears in bold text in the Design Runs view.
- **Run Multiple Strategies**—Invokes the Run Multiple Strategies dialog box to create and launch multiple runs. For more information, see [“Creating Multiple Synthesis Runs”](#) or [“Creating Multiple Implementation Runs.”](#)
- **Run Bitgen**—Invokes the Run Bitgen dialog box to create a bitstream. This command is only available for completed implementation runs.
- **Export to Spreadsheet**—Creates a Microsoft Excel format spreadsheet file containing the entire Design Runs table view.

Viewing and Modifying Run Properties

Each run has a variety of run properties that can be viewed or modified. Modification of most Run Properties is only possible prior to launching the run. Once the run has been launched, select the **Reset Run** popup command to view and edit most Run Properties.

Viewing the Run General Properties

Select the Run, and then select the **General** tab in the Run Properties view.

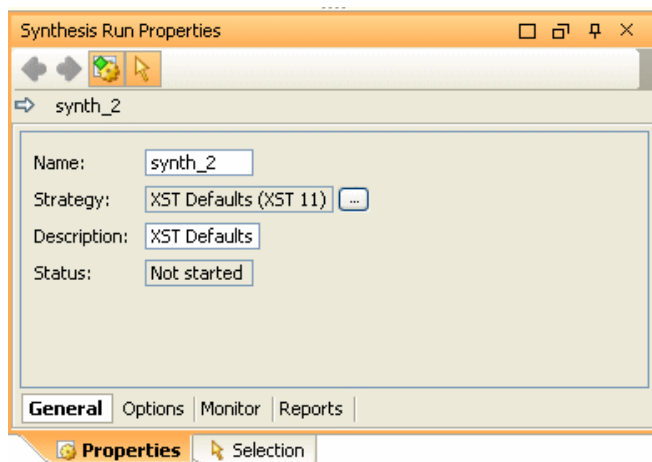


Figure 7-16: Setting Synthesis General Properties

View or edit the information found in the Synthesis General Run Properties dialog box:

- **Name**—Defines the Run name.
- **Strategy**—Invokes the Strategy chooser to select a Run Strategy.
- **Description**—Defines the Run description.
- **Status**—Displays the status of the Run.

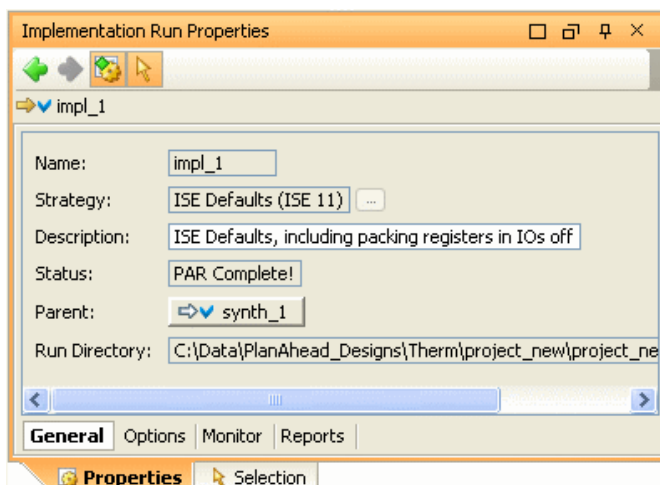


Figure 7-17: Setting Implementation General Properties

View or edit the information found in the Implementation General Run Properties dialog box:

- **Name**—Defines the Run name.
- **Strategy**—Invokes the Strategy browser to select a Run Strategy.
- **Description**—Defines the Run description.
- **Status**—Displays the status of the Run.
- **Parent**—Displays the synthesis Netlist or Pblock being implemented.
- **Run Directory**—Displays the location that the run data exists or will be created in.

Viewing the Run Strategy Options

The command line options that are defined in the Strategy can be viewed and modified in the Run Properties Options dialog box.

Select the Run, and then select the **Options** tab in the Run Properties view.

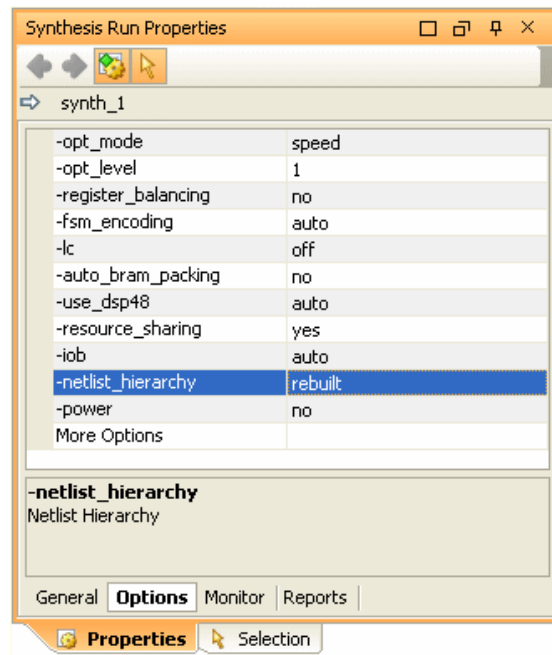


Figure 7-18: Setting Synthesis Strategy Options

All of the command line options and their preset values are displayed in the view. A description of the command option intent can be displayed by selecting any of the options.

Selecting an option enables a pulldown menu on the right side to view the available values for the option.

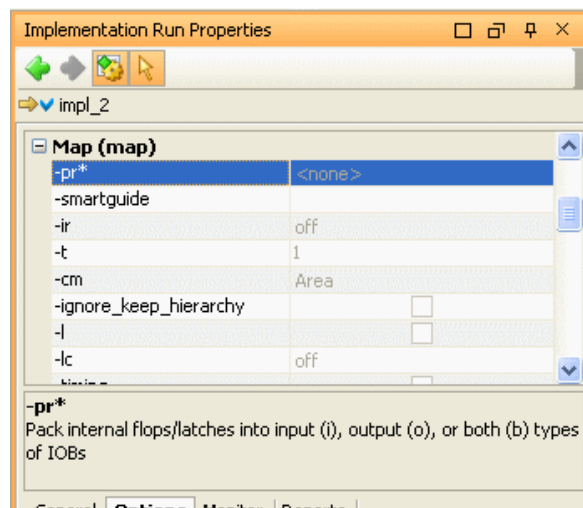


Figure 7-19: Setting Implementation Strategy Options

Overriding ISE Command Options Set in a Strategy

The Options tab enables you modify Strategy options.

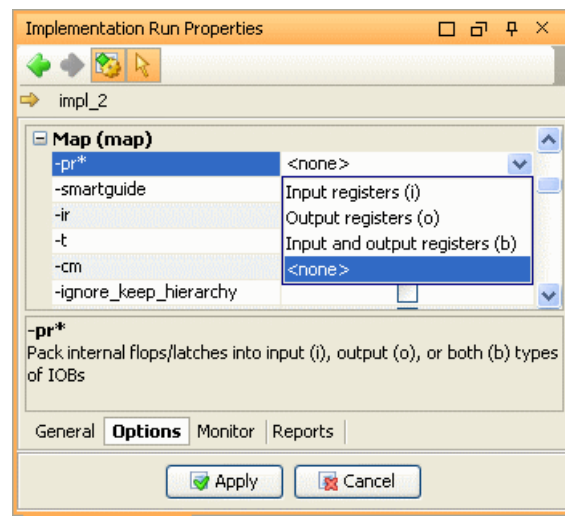


Figure 7-20: Run Properties View: Options Tab

Using the popup commands, you can override (edit) the Strategy for the selected run and save (or save as) the options to a new User Defined Strategy. Additional command options that are not displayed can be entered in the More Options field under the desired command.

Click the **Apply** toolbar button to apply all changes.

Any values modified receive an "*" next to the option indicating that the default Strategy value has been changed. The Strategy field for the Run in the Design Runs view also receives an "*" indicating that the default Strategy has been changed.

Once a Strategy is launched (as described in the next section), the Strategy options can no longer be modified. To edit an option, you will need to reset the run and then edit the option. See ["Resetting Runs."](#)

Monitoring Run Status

Select the Run, and then select the **Monitor** tab of the Run Properties view to view the status of the run while it is running. The view displays the same "standard out" command status logs that displays when at the command line.

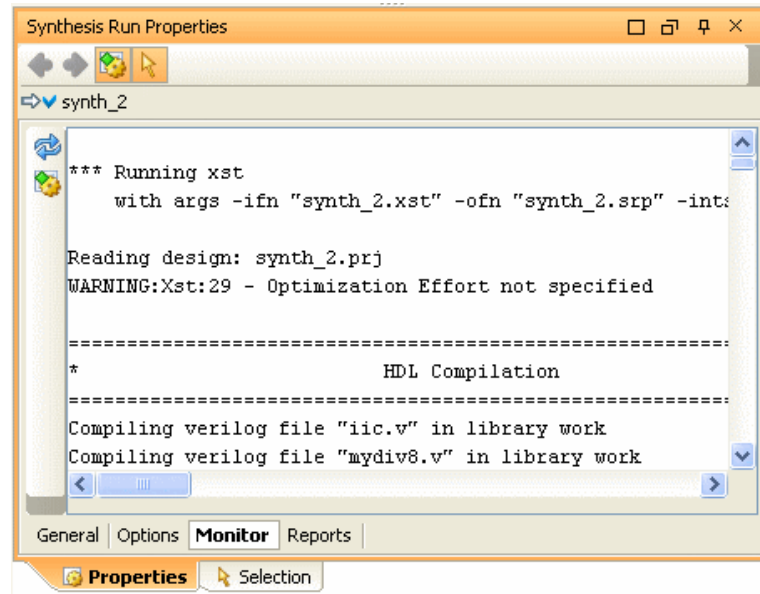


Figure 7-21: Monitoring Run Status

The Monitor view will continue to update as the commands are being run. The scroll bar can be used to browse through the command log reports. Select the **Automatically update the contents of this view** button to stop the active reporting. This may enable you to better scroll and read results while the command is running.

Viewing Report Files

Report files generated by the ISE tool can be viewed from with PlanAhead. Select the Run, and then select the **Reports** tab in the Run Properties view to display the list of available report files.

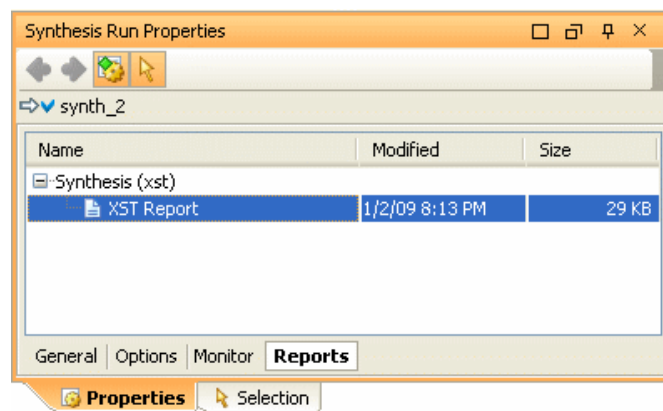


Figure 7-22: Selecting Report Files to View

Selecting any of the available report files will display it in the Workspace view.

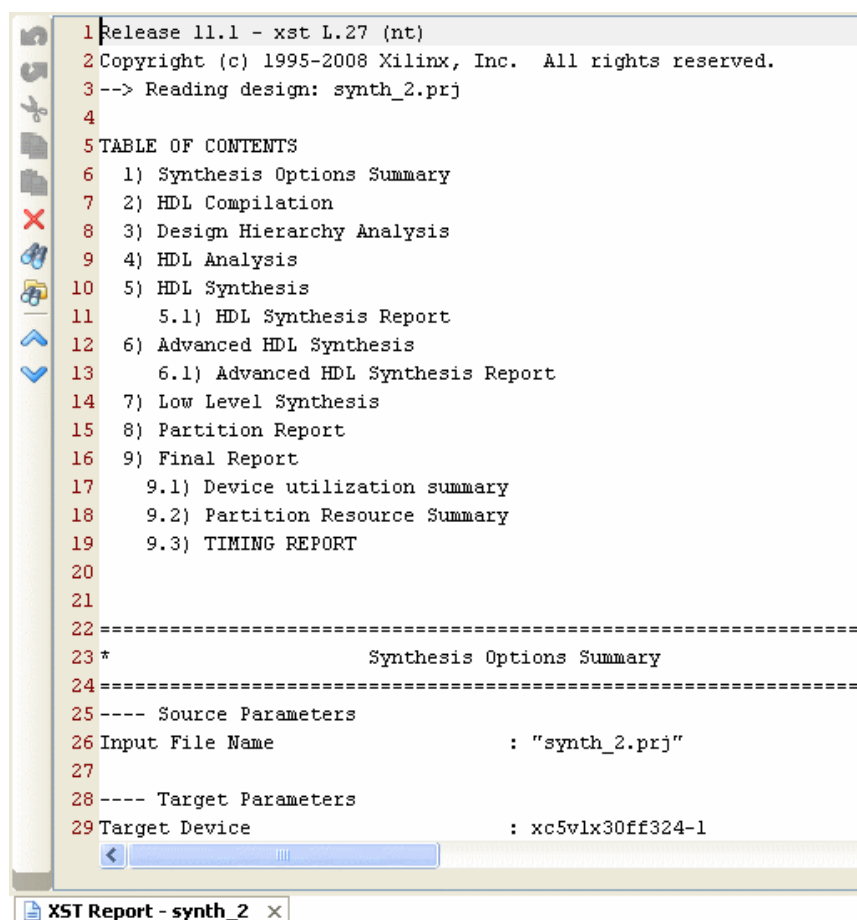


Figure 7-23: Viewing Report Files

The scroll bar can be used to browse the report file. Select the Find or Find in Files Toolbar buttons to search for specific text. Use the Go to the beginning or Go to the End toolbar buttons to scroll to the beginning or end of the file.

Managing Runs

Launching Existing Runs

The Launch Runs command is used to launch existing Runs in the Design Runs view. Launching Runs can be performed on Runs in any state, including completed Runs. The Launch Selected Runs dialog box is first displayed to set launch options.

1. In the Design Runs view, select one or more runs. Use **Shift**+click or **Ctrl**+click for multiple selections.
2. Select one of the following commands:
 - ♦ Select the **Launch Runs** popup command.
 - ♦ Select the **Launch selected runs** Design Runs toolbar button.



Figure 7-24: Launch Selected Runs Toolbar Button

The Launch Selected Runs dialog box appears.

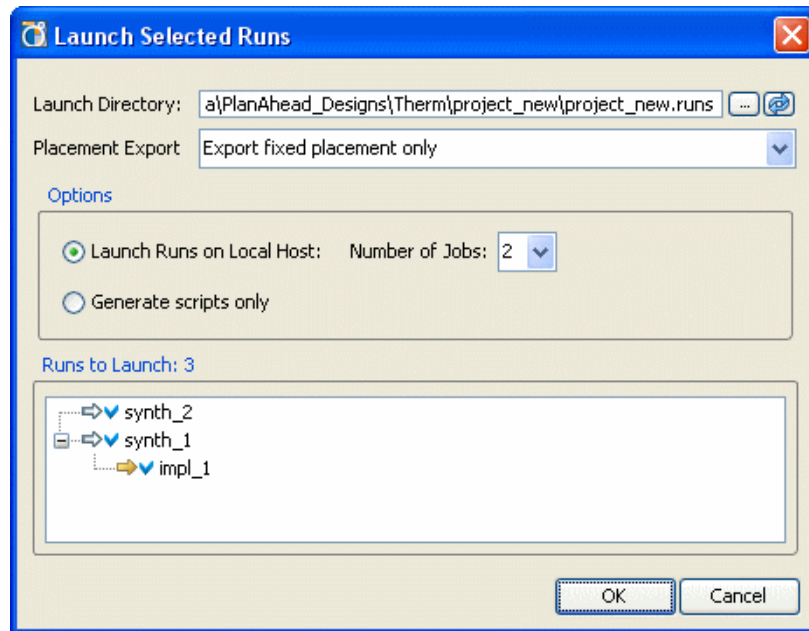


Figure 7-25: Launching Existing Runs

The Launch Selected Runs dialog contains the following options:

- ◆ **Launch Directory**—Specify a location to create and store the synthesis run data.
Note: Defining any non-default location outside of the project directory structure makes the project non-portable as absolute paths are written into the project files.
 - ◆ **Placement Export**—Specify to export placement LOC constraints for only user assigned “fixed” placement, or for all “fixed” and “unfixed” placement constraints from an ISE implementation run.
 - ◆ **Launch Runs on Local Host**—Select this option to launch the Run on the local machine processor.
 - **Number of Jobs**—Define the number of local processors to use for Runs. This option is only used when launching multiple runs simultaneously. Individual runs will be launched on each processor. No multi-threaded processors are used with this option.
 - ◆ **Launch Runs on Remote Hosts** (Linux only)—Select this option to use remote hosts to launch job or jobs.
 - **Configure Hosts**—Select this option to configure remote hosts. For more information, see [“Executing Runs on Multiple Linux Hosts.”](#)
 - ◆ **Generate scripts only**—Select this option to export and create the run directory and run script, but not to launch the run from PlanAhead. The script can be run at a later time outside of the PlanAhead environment.
3. Click **OK** to create the Run with the selected launch options.
 4. If the selected Runs are in a state other than “Not Started, you are prompted to first reset the runs prior to launching them.

Resetting Runs

The Reset Runs command is used to remove the results of the selected runs. You are prompted to remove the run data from disk, which is advisable. The status of the Run is set back to *Not Started*.

1. Select one or more runs in the Design Runs view. Use **Shift**+click or **Ctrl**+click for multiple selections.
2. Select the **Reset Runs** popup command.

The Reset Runs confirming dialog box will appear to prompt you to remove all implementation data from disk for the selected runs.

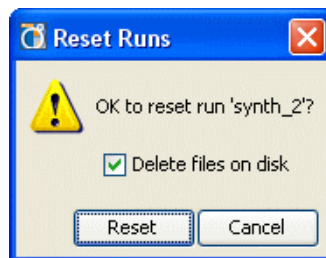


Figure 7-26: Resetting Runs

3. In the confirming dialog box, click **Reset**.
If any ISE processes are currently running or queued, you are prompted to stop them.
4. Click **Yes** to proceed.

The status for the selected runs is reset.

Deleting Runs

The Delete command is used to remove selected Runs from the Design Runs view and to remove their associated data from disk. You are prompted to confirm the deletion of the selected runs.

1. Select one or more runs in the Design Runs view. Use **Shift**+click or **Ctrl**+click for multiple selections.
2. Select one of the following commands:
 - ♦ Select the **Delete** main toolbar button.



Figure 7-27: Deleting Selected Runs Toolbar Button

- ♦ Select the **Delete** popup menu command in the Design Runs view
- ♦ Select the **Edit > Delete** command
- ♦ Press the **Delete** key

Importing Run Results

Importing Synthesis Run Results

PlanAhead enables you to import the results of a synthesis run for further analysis, I/O pin planning, floorplanning and implementation. Once the synthesis run has completed, a dialog box is displayed prompting you to import the run. For more information, see [“Running Synthesis.”](#)

During the import, you can elect to import the netlist for logical netlist exploration only, or create a Floorplan, which enables a wide range of design analysis, floorplanning and implementation options.

Once a synthesis run is complete, you can import the resulting netlist as follows:

1. Select an synthesis run in the Design Runs view.
2. Select one of the following commands:
 - ♦ Select **Import Run** from the popup menu in the Design Runs view.
 - ♦ Select the **Import Run** toolbar button in the Design Runs view.



Figure 7-28: **Import Run Toolbar Button**

Note: Double-clicking on any completed run in the Design Runs view will invoke the import Run dialog on that run.

The Import Synthesis Results wizard is invoked.

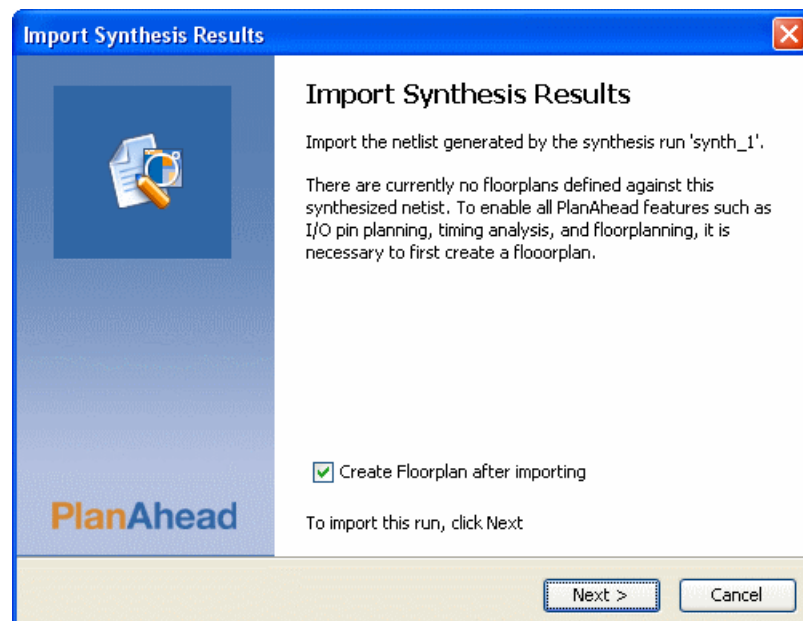


Figure 7-29: **Import Synthesis Results**

The Create a Floorplan after importing option is selected by default. Deselect the option if you do not wish to create a floorplan.

Creating a Floorplan with Synthesis Results

The Choose a Part and a Floorplan Name dialog box of the Import Synthesis Results wizard is now invoked.

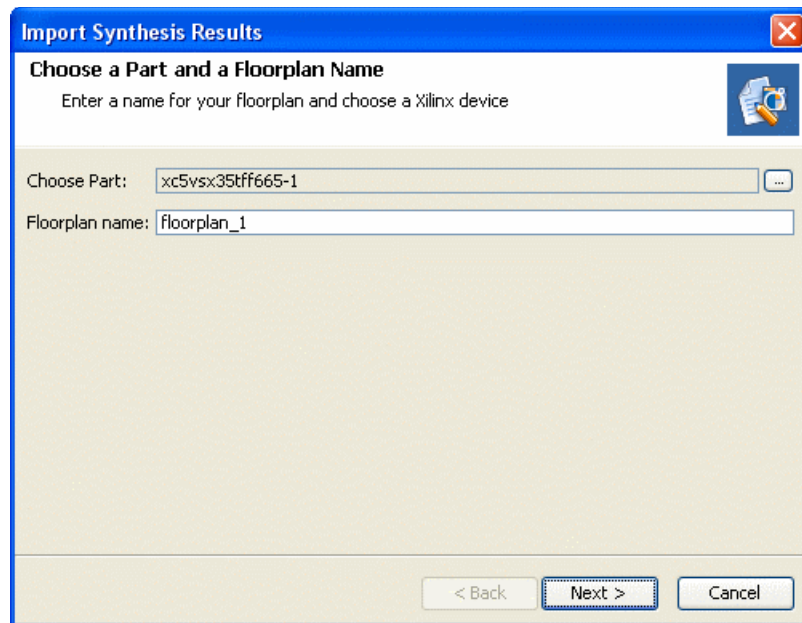


Figure 7-30: Import Synthesis Results: Choose Part and Floorplan Name

3. Set the following editable options in the Floorplan Name dialog box.
 - ◆ **Choose Part**—Use the files browser to invoke the Select Part chooser which enables you to select a desired device.
 - ◆ **Floorplan name**—Enter the desired name for the floorplan.
4. Click **Next** to continue.

The Import Constraints dialog box of the New Project wizard is now invoked.

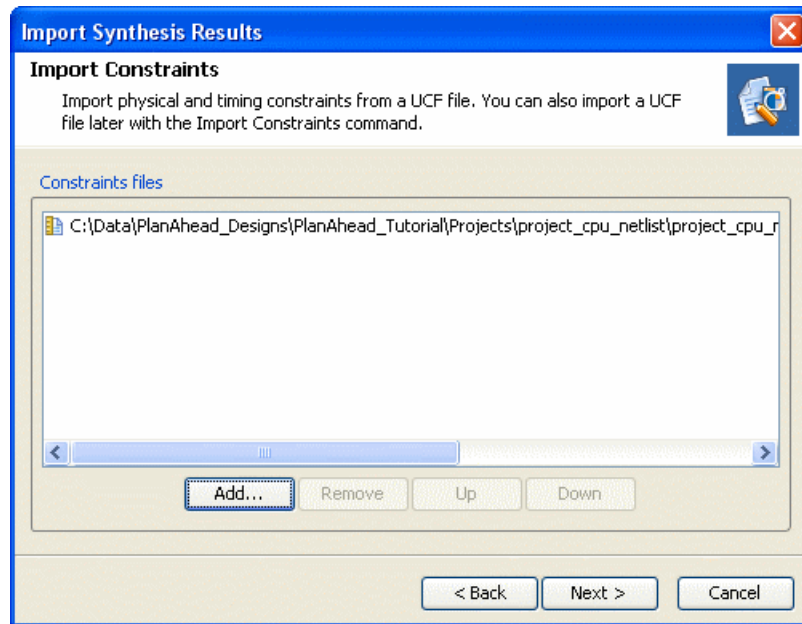


Figure 7-31: New Floorplan Wizard: Import Constraints

5. Click the **Add** button to locate and select *top-level* UCF and NCF constraint files for import. You can arrange the order in which to import these files by selecting them and using the Up or Down buttons. Files can be removed from the list by using the Remove button.

If module level constraints are being used, do not include them here. For more information on importing module-level constraints, see [“Importing Module-Level Constraints.”](#)

6. Click **Next** to continue.

The UCF files are imported into PlanAhead. This may take a few moments.

The Import Synthesis Results Summary dialog box is now displayed.

7. Click **Finish** to proceed.

The PlanAhead environment displays with Synthesis run results imported and a new floorplan open, if one was created.

Importing Implementation Run Results

Implementation run results can be imported into PlanAhead for further analysis or floorplanning. The placement is imported and displayed in the form of “unfixed” LOC constraints. You can visualize where the ISE implementation tools placed various logic objects. The Trace timing results are also imported to enable visualization and exploration of the timing path logic. For more information, see [Chapter 8, “Analyzing the Design.”](#)

Once a implementation run is complete, you can import ISE placement and timing results as follows:

1. Select an implementation run in the Design Runs view.
2. Select one of the following commands:
 - ♦ Select **Import Run** from the popup menu.
 - ♦ Select the **Import Run** toolbar button in the Design Runs view.



Figure 7-32: **Import Run Toolbar Button**

Note: Double-clicking on any completed run in the Design Runs view will invoke the import Run dialog on that run.

The Implementation Run placement and timing results are displayed. The UCF constraints that were used to generate the run are imported as well. If modifications are made to the constraints in PlanAhead, you are prompted to save them as a floorplan.

Multiple runs results can be imported. A tab is displayed with the run name at the top of the PlanAhead environment for each run imported. Selecting the tab will display the associated run results data.

Importing Run Results for a Floorplan

The run results associated with a floorplan are imported as described above. The only difference is that only one run result per floorplan can be displayed at a time. A tab is displayed with the floorplan name at the top of the PlanAhead environment. Each time a floorplan run result is imported, the results are displayed with the same Floorplan view layout tab.

Running Bitgen on an Implementation Run

Once a run has been completed, the ISE Bitgen command can be run on the results to create the bitstream data. To do so:

1. Select the desired completed run in the Design Runs view.
2. Select the **Run BitGen** command from the popup menu.

The run Bitgen dialog box will appear.

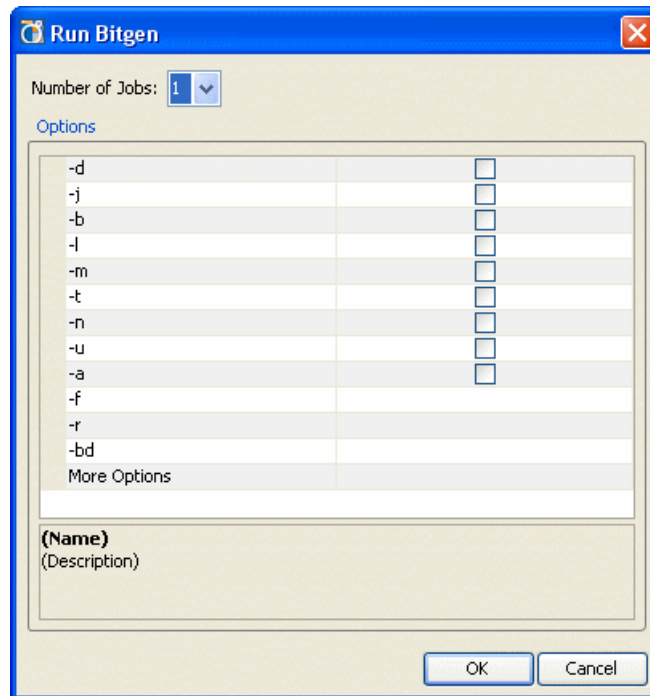


Figure 7-33: Setting Bitgen Options

The Bitgen command options can be set prior to running the command. Selecting an option will display a description of the option intent in the dialog box. A pulldown menu of the available options values is presented on the right side

3. Click **OK** to start the bitgen command

The command status can be viewed in the Run Properties Monitor tab and the Bitgen report file can be viewed in the Run Properties Reports tab.

The resulting bit file is generated in the Run directory.

Creating Strategies

Creating Synthesis and Implementation Strategies

A strategy is a set of command options for each ISE implementation command. PlanAhead is shipped with several commonly used Strategies that have been used extensively on internal benchmarks with great success. The option settings for these strategies are not editable. You can copy and modify these Strategies to create your own.

Strategies are Tool and version specific. Each major release of ISE have version-specific command line options, which PlanAhead can support. Select or copy a strategy from a specific tool and version category as shown below.

To review, copy and modify strategies:

1. Select **Tools > Options > Strategies**.

The Strategies dialog box will appear. It contains a list of pre-defined strategies.

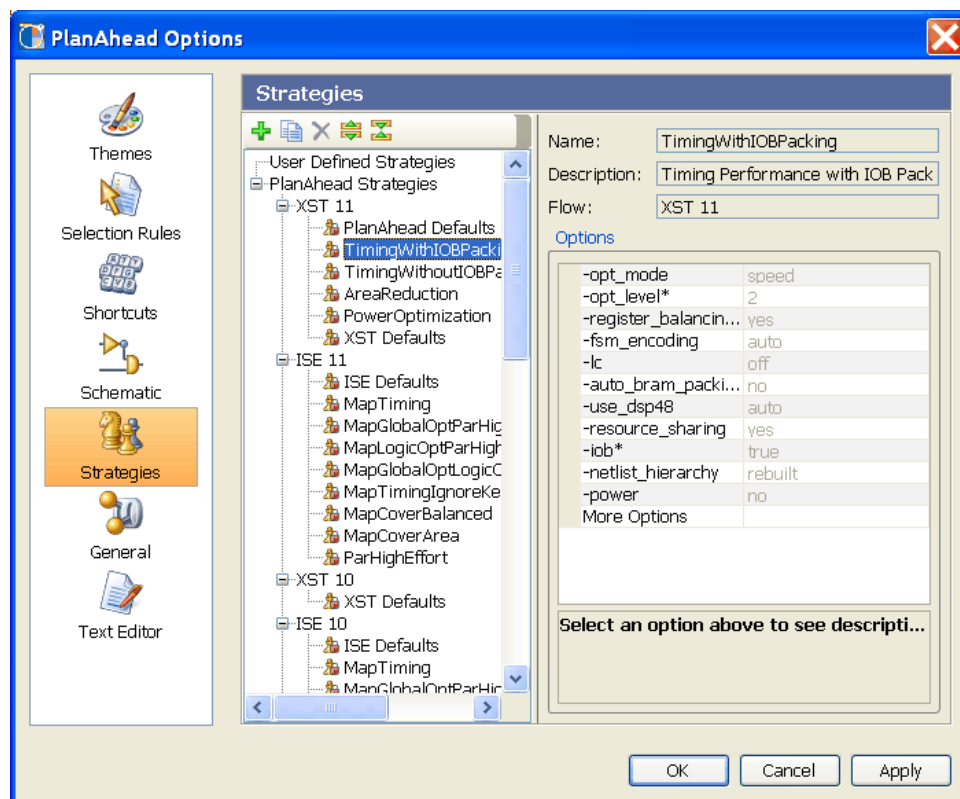


Figure 7-34: PlanAhead Options: Strategies

2. Copy the supplied PlanAhead Strategies to the User Defined Strategies area for modification by using one of the following commands:
 - ◆ Select the **Create a copy of this strategy** Strategies toolbar button.



Figure 7-35: Create a Copy of this Strategy Toolbar Button

- ◆ Select the **Copy Strategy** popup command.
A copy of the strategy will be made in the User Defined Strategies area allowing modification of command option values on the right side of the dialog box. For more information about the strategy file, see [“Strategy Files \(<strategyname>.psg\).”](#)
3. Edit the definable option in the Strategies dialog box as follows:
 - ◆ **Name**—Enter a Strategy name to assign to a Run.
 - ◆ **Description**—Enter the Strategy description which will be displayed in the Design Run results table.
 - ◆ **Flow**—Enter the tool and version the strategy is intended for.
 4. Click a command option to view the option description at the bottom of the dialog box.

5. Modify command options by clicking in the command option area (to the right), and selecting an option from the popup menu. All available command option settings are displayed in the popup menu.

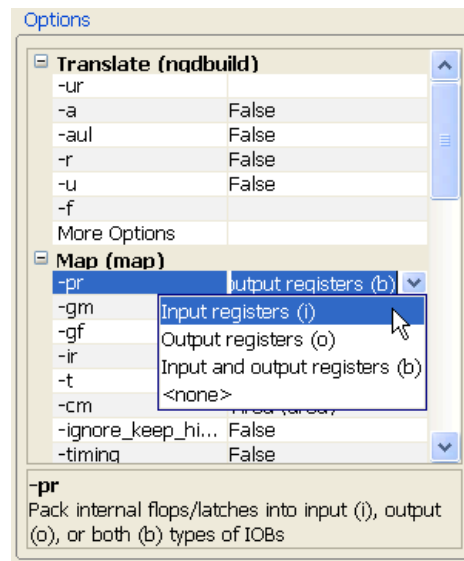


Figure 7-36: Command Options and Description

6. Click **Apply** and **OK** to save the new strategy.

The new strategy appears as a User Defined Strategy and can be used for Design runs, as described in [“Using the Design Runs View.”](#)

For more information about saved Strategies, see [“Outputs for Environment Defaults.”](#)

Creating Common Group Strategies

Design groups that want to create and use group-wide custom strategies can copy any User Defined Strategy to the following directory:

- `<InstallDir>/strategies`

Executing Runs on Multiple Linux Hosts

PlanAhead currently is shipping functionality to allow parallel execution of Runs on multiple Linux hosts. This is accomplished with simplified versions of more robust load-sharing software, such as Sun’s Grid Engine, and LSF. Job submission algorithms are implemented using a greedy, round-robin style with Tcl pipes with Secure Shell (SSH).

Limitations

The limitations are as follows:

- Host execution is performed with SSH, a service provided by Linux operating system, and not PlanAhead. In order for this to work, you must configure SSH so that you are not prompted for a password each time you log in to a remote machine. If you have not configured key-agent forwarding for passwordless SSH, or if you have configured SSH and you are prompted for a password, see [Appendix C, “Configuring SSH Without Password Prompting.”](#)

- Linux-to-Linux hosting is the only supported platform because of security and lack of remote-shell capabilities of windows systems.
- ISE tool installation is assumed to be available from any login shell, which means that \$XILINX and \$PATH are configured correctly in your .cshrc / .bashrc setup scripts. If you can log into a remote machine and enter “map -help” without sourcing any other scripts, this flow will work. If you do not have ISE set up upon login (.cshrc or .bashrc), you can use the “Run pre-launch script” option to pass an environment setup script to be run prior to all jobs.
- PlanAhead installation must be visible from the mounted file systems on remote machines. If the PlanAhead installation is stored on a local disk on your own machine, it will not be visible from remote machines.
- PlanAhead project files (.ppr) and directories (.data and .runs) must also be visible from the mounted file systems on remote machines. If the design data is saved to a local disk, it will not be visible from remote machines.

Configuring Remote Hosts (Linux Only)

After you have configured SSH, as described in APPENDIX C, PlanAhead enables runs to be launched using remote servers. In order to do so, they must first be configured.

1. To configure a remote host select one of the following commands:
 - ♦ Select **Tools > Options > Remote Hosts**.
 - ♦ Select the **Configure Hosts** button within the Run Synthesis > Launch Runs options dialog box.
 - ♦ Select the **Configure Hosts** button within the Run Implementation > Launch Runs options dialog box.
 - ♦ Select the **Configure Hosts** button within the Launch Selected Runs options dialog box.

The Remote Hosts dialog box displays.

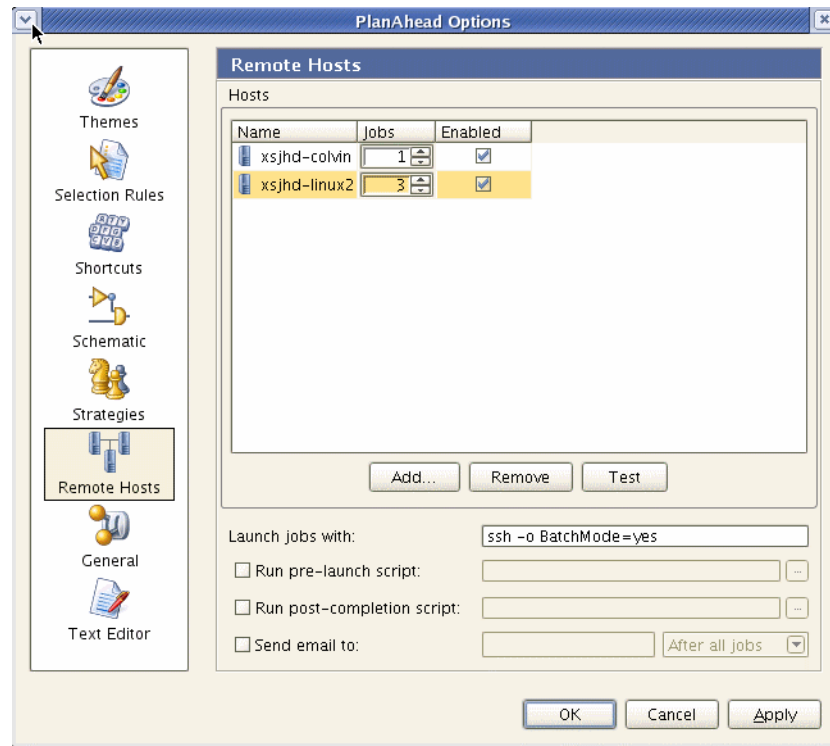


Figure 7-37: Configuring Remote Hosts

2. Click the **Add** button to enter the names of remote servers
3. Toggle the **Jobs** option to specify how many processors on the remote machine to use. Individual runs are launched on each processor. No multi-threading of processors is used.
4. Toggle the **Enable** option to specify whether to use the server. You can use this field when launching runs to specify which servers to use for the selected runs to be launched.
5. Optionally, modify the Launch jobs with field to change the remote access command. The default is `ssh`.

Note: Use extreme caution when modifying this field. For example, removing 'BatchMode =yes' may cause the process to hang because the shell incorrectly prompts for an interactive password.

6. Optionally, click the **Run pre-launch script** button and define a script to run prior to launching the runs. Use this option to pass an environment setup script if you do not have ISE set up upon login (`.cshrc` or `.bashrc`).
7. Optionally, click the **Run post-completion script** button and define a custom script to run after the run completes.
8. Optionally, click the **Send email to** button and enter an email address to send notification when the run completes.
9. Select one or more hosts, and select the **Test** button to verify that the server is available and the configuration is set up properly.

Note: It is highly recommended that you test each host to ensure proper set up.

10. Select the **Remove** button to delete selected remote hosts.
11. Click **OK** to accept the Remote Host configuration settings.

Interfacing with ISE Outside of PlanAhead

The PlanAhead software enables you to selectively export files required for external ISE software implementation. If you are using PlanAhead for design implementation, interfacing with ISE for external implementation is not necessary, as described elsewhere in this chapter.

PlanAhead also enables the creation of a Project based on existing command line based implementation results. For more information, see [“Creating an Implemented Design based Project.”](#)

Exporting Constraints

Exporting floorplan constraints to ISE consists of exporting a UCF physical constraints file for the entire design or for individual Pblocks.

To export the floorplan constraints:

1. Click the desired Floorplan tab.
2. Select **File > Export Constraints**.

The Export Constraints dialog box is displayed.

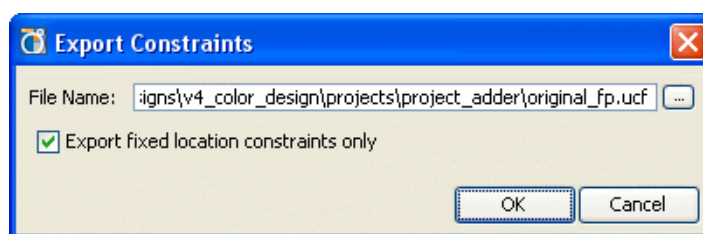


Figure 7-38: Export Constraints Dialog Box

3. View and edit the definable options in the Export Constraints dialog box:
 - ♦ **File name**—Enter the file name and location to create the UCF format constraints file.
 - ♦ **Export fixed location constraints only**—Select this option to export only the user assigned “fixed” placement LOC constraints or uncheck it to export all of the fixed and unfixed placement constraints imported from ISE.
4. Click **OK** to export the constraints.

PlanAhead will create the designated top-level UCF format constraint file in the export directory. This file can be used as input for custom ISE implementation scripts.

For more information about the exported files, see [“Outputs for ISE Implementation.”](#)

Exporting Netlist

Exporting the PlanAhead Netlist to ISE consists of exporting a single EDIF format netlist file for the entire design or for individual Pblocks.

To export the design netlist:

1. Click the Floorplan tab.
2. Select **File > Export Netlist**.

The Export Netlist dialog box is displayed.

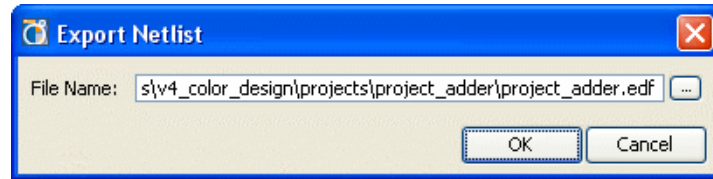


Figure 7-39: Export Netlist Dialog Box

3. View and edit the definable option in the Export Netlist dialog box:
 - ♦ **File name**—Enter the file name and location to create the EDIF format netlist file.
4. Click **OK** to export the netlist.

For more information about the exported files, see [“Outputs for ISE Implementation.”](#)

Exporting Pblocks for ISE Implementation

PlanAhead has the unique ability to export Pblock-level files for implementation. These Pblocks consist of logic from anywhere in the logic hierarchy. Exporting a Pblock creates an EDIF netlist and UCF physical constraints file for each Pblock selected for export.

To export Pblocks to EDIF and UCF:

1. Select one or more Pblocks.
2. Select **File > Export Pblocks**.

This Export Pblocks wizard is displayed.

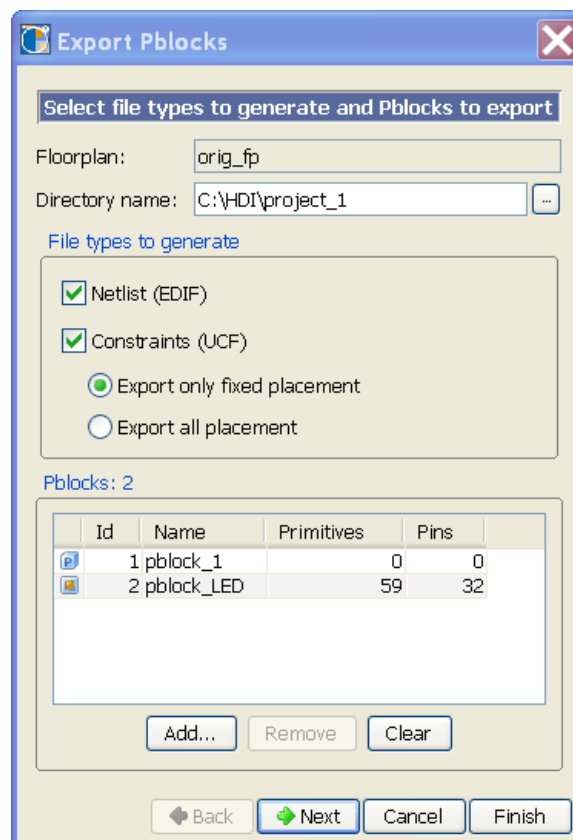


Figure 7-40: Export Pblocks Wizard

3. View and edit the definable options in the Export Pblocks dialog box:
 - ◆ **Directory name**—Enter the directory name or use the file browser to select a directory to export the files. A subdirectory called `<pblockname>_CV` is created for each exported Pblocks. In order to help keep track of the various EDIF and UCF files associated with a typical PlanAhead design scenario, it is good practice to specify a unique directory name for each ISE attempt. The export directory will be used to seed the “Import Placement” and “Import TRCE Results” command file browsers.
 - ◆ **File types to generate**
 - **Netlist (EDIF)**—Select this option to export the netlist.
 - **Constraints (UCF)**—Select this option to export all or only fixed placement constraints.
 - ◆ **Pblocks**—Lists the Pblocks selected for export.
4. Click the **Add** and **Clear** buttons to select and remove Pblocks from the export list, respectively.
5. Click **Next** or **Finish** to continue.
6. If Next is selected, the Export Pblocks Summary dialog box displays Pblock export selections.
7. Click **Finish** to perform the export.

PlanAhead will always create a separate EDIF and UCF files for each of the exported Pblocks named `<pblockname>_CV.edn` and `<pblockname>_CV.ucf`. A `<pblockname>_CV` directory will be created for each exported Pblock containing the Pblock specific files.

Importing ISE Implementation Results

Creating an Implemented Design based Project

Creating a project that imports implementation results is described in [“Creating a Project with ISE Placement and Timing Results.”](#)

Importing Placement Results into Existing Project

Placement results from ISE can be imported into PlanAhead. Placement constraints will be assigned for all placed logic objects. Placement results can be imported for the top level of the design or for individual Pblocks.

XDL format files are used to import placement results. PlanAhead will automatically run the ISE XDL command to create the file from your `<design_name>routed.ncd` file.

To import placement results:

1. Select **File > Import Placement**.

The Import Placement dialog box will display.

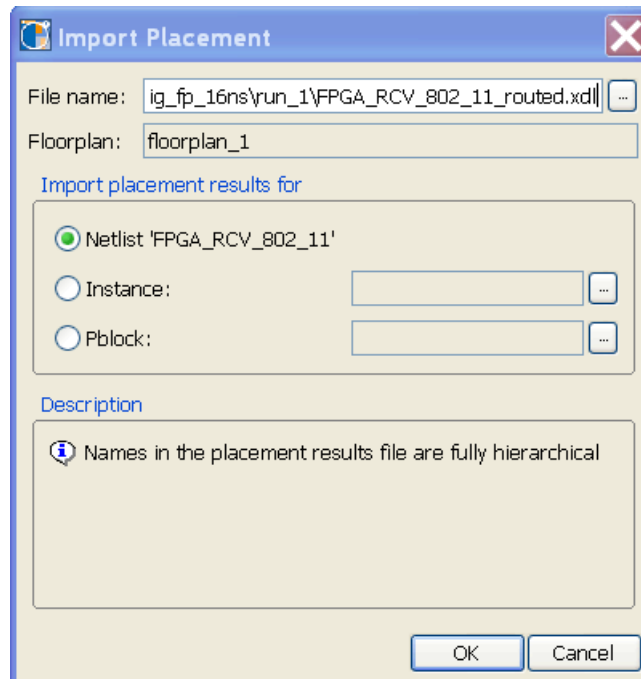


Figure 7-41: Import Placement Dialog Box

2. View and edit the definable fields are available in the Import Placement dialog box:
 - ♦ **File Name**—Click the browse button to locate and select either the XDL format file or the `<design_name_routed>.ncd` file. If the NCD format file is selected, PlanAhead will automatically convert it to XDL format and import the XDL result file.
 - ♦ **Import placement results for**—Selects the level of the design to import placement for. This command is seeded with pre-selected objects, so carefully select the proper level of the design.
 - **Netlist '*<netlist name>*'**—Select this option when importing a top-level XDL. This is typically the case when running ISE on the top-level design.
 - **Instance**—Select this option to import placement results for a selected logic instance. The dialog box can be seeded with a pre-selected instance prior to running the command. This option is useful when importing block-level placement constraints.
 - **Pblock**—Select this option to import results from a previously implemented Pblocks.
3. Click **OK** to import the placement results.

The results of importing placement are shown below. Net connection and traced paths will now display to the placement constraint location, instead of the Pblock center.

Importing ISE TRCE Timing Results into an Existing Project

TimeAhead also has the ability to import the `.twx` and `.twr` format timing reports generated by the Xilinx `trce` command. Once imported, all the signal tracing and viewing capabilities described in this chapter are available with the TimeAhead environment.

To import timing results:

1. Select **File > Import TRCE Results**.

The Import TRCE Results dialog box appears.

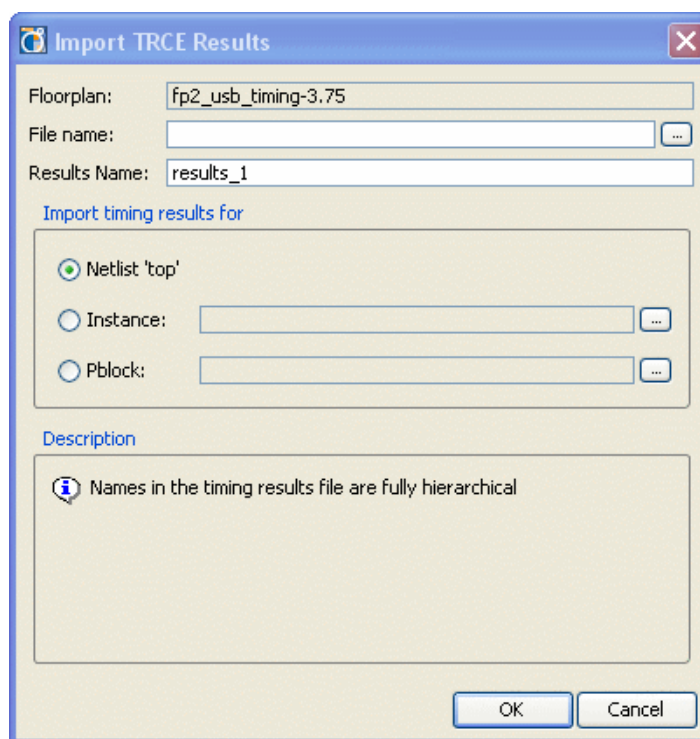


Figure 7-42: Import TRCE Results Dialog Box

2. View and edit the definable fields in the Import TRCE Results dialog box:
 - ◆ **File Name**—Define a *trce* format .twx or .twr file name for PlanAhead to import results.
 - ◆ **Results Name**—Define a name to appear on the results tab in the Timing Results view.
 - ◆ **Import timing results for**—Select the level of the design to import. This command is seeded with pre-selected objects, so carefully select the proper level of the design.
 - **Netlist '<netlist_name>'**—Select this option to import results for a top-level Trce result.
 - **Instance**—Select this option to import results for a instance. Click the browse button to select a specific instance.
 - **Pblock**—Select this option to import results for a Pblock. Click the browse button to select a specific Pblock.
3. Click **OK** to import the timing results.

The TRCE results are displayed within the TimeAhead environment.

Name	Type	Slack	From	To	Total Delay	Logic Delay	Net %	Stages
Constrained Paths (240)								
TS_usbClk = PERIOD TIMEGRP "usbClk" 3.9 ns HIGH 50%; (30)								
Path 1	Setup	-0.265	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[7]	4.014	1.263	68.535	6
Path 2	Setup	-0.223	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[9]	3.948	1.171	70.339	5
Path 3	Setup	-0.221	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[9]	3.967	1.188	70.053	5
Path 4	Setup	-0.098	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[23]	3.853	1.282	66.727	6
Path 5	Setup	-0.086	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine0/u4/dout[23]	3.811	1.126	70.454	6
Path 6	Setup	-0.050	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/inta_msk[4]	3.822	1.391	63.605	4
Path 7	Setup	-0.050	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/inta_msk[5]	3.822	1.391	63.605	4
Path 8	Setup	-0.050	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/inta_msk[7]	3.822	1.391	63.605	4
Path 9	Setup	-0.050	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/inta_msk[6]	3.822	1.391	63.605	4
Path 10	Setup	-0.039	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[7]	3.767	1.258	66.605	6
Path 11	Setup	-0.023	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_...	usbEngine1/u4/dout[25]	3.776	1.191	68.459	5

Figure 7-43: TRCE Timing Results

All of the path selection, highlighting and tracing capabilities can now be explored using the TimeAhead interface.

Slack values appear red for paths with negative slack values.

The Timing Results can be sorted by clicking any of the column headers. You can sort by a second column by pressing the **Ctrl** key and clicking a second column header. Add as many sort criteria as necessary to refine the list order.

Multiple Timing Results can be displayed for a Floorplan. Each result from Trce or TimeAhead will receive a tab at the bottom of the report, as shown above.

Analyzing the Design

This chapter describes the many design analysis features of the Floorplan environment available in the PlanAhead™ software. There are additional analysis features described in [Chapter 5, “I/O Pin Planning,”](#) [Chapter 6, “Creating and Analyzing the RTL Design,”](#) and [Chapter 9, “Analyzing Implementation Results.”](#)

Most of these features can be used prior to implementing the design. They are intended to identify any potential design issues prior to running the implementation tools.

There are also a wide range of features useful for analyzing an implemented design. PlanAhead enables you to import the placement and timing results from ISE® implementation runs for further analysis and floorplanning.

This chapter contains the following sections:

- [“Using the Floorplanning Environment”](#)
- [“Analyzing the I/O Pinout and Clock Logic”](#)
- [“Analyzing the RTL Design”](#)
- [“Analyzing the Synthesized Design”](#)

Using the Floorplanning Environment

Using the Device View

The Device view displays all of the FPGA device resources including the logic fabric, clock regions, I/O pads, BUFGs, DCMs, Pblocks, instance locations, and net connectivity. The locations on the device where specific logic can be assigned are referred to as Sites.



Figure 8-1: Device View

The amount of logic object detail displayed depends on the zoom level selected. The more the zoom level is increased, the more logic object detail is displayed. There are many self explanatory zoom level related commands contained in the popup and toolbar menus in the Device view. The Device view also has scroll bars and dynamic pan capabilities to pan the viewable area of the Device.

Each object displayed in the Device view can be identified by dragging the cursor over it and reading the tool tip displayed. They also have object properties that can be viewed in the Properties view when selected. Each selected object has popup menu commands specific to the type of object selected. Sometimes the desired commands for selected objects are only available if the cursor is moved to another more appropriate view for the command, such as the Netlist view. Use **Edit > Find** to search for specific logic object sites.

A dynamic cursor is utilized within the Device view. The appearance of the cursor will change depending on the activity being performed. The dynamic cursor also changes to indicate when an illegal logic resource assignment is being attempted. For more information, see [“Understanding the Context Sensitive Cursor.”](#)

There are several icons in the upper left corner toolbar of the Device view. These view specific toolbar commands are covered in [“Manipulating Views using the View Banner Commands.”](#)

The Device view is also used during the I/O pin planning process. For more information about using the Device view during pin assignment, see [Chapter 5, “I/O Pin Planning.”](#)

Viewing Device Resources

PlanAhead displays the various resources contained in the selected device. The level of detail at which the Device resources are displayed depends on the active zoom level within the Device view. Graphical sites are displayed and available for all of the device specific FPGA resources.

The I/O pads and clock objects are displayed around the periphery and/or down the center of the device. I/O banks are displayed as thin color shaded rectangles just outside the row of I/O pads. Available I/O bank sites are indicated with colored filled I/O bank rectangles. Some devices have unbonded I/O banks which are displayed with empty I/O bank rectangles. The I/O clock pads are shown as filled-in rectangles. All clock resources, such as BUFG, BUFRs and BUFGPs, are also shown distinctly in the Device view. Selecting an I/O bank will display all of the available device resources in the I/O Bank Properties view.

The interior of the device is broken up into smaller rectangles called tiles. These tiles consist of placement sites for the different types of logic primitives for the architecture being used. Virtex®-4, Virtex-5 and Virtex-6 devices have many different types of logic sites. Each site in the Device view can be identified by a tool tip when you hover the cursor over a logic site.

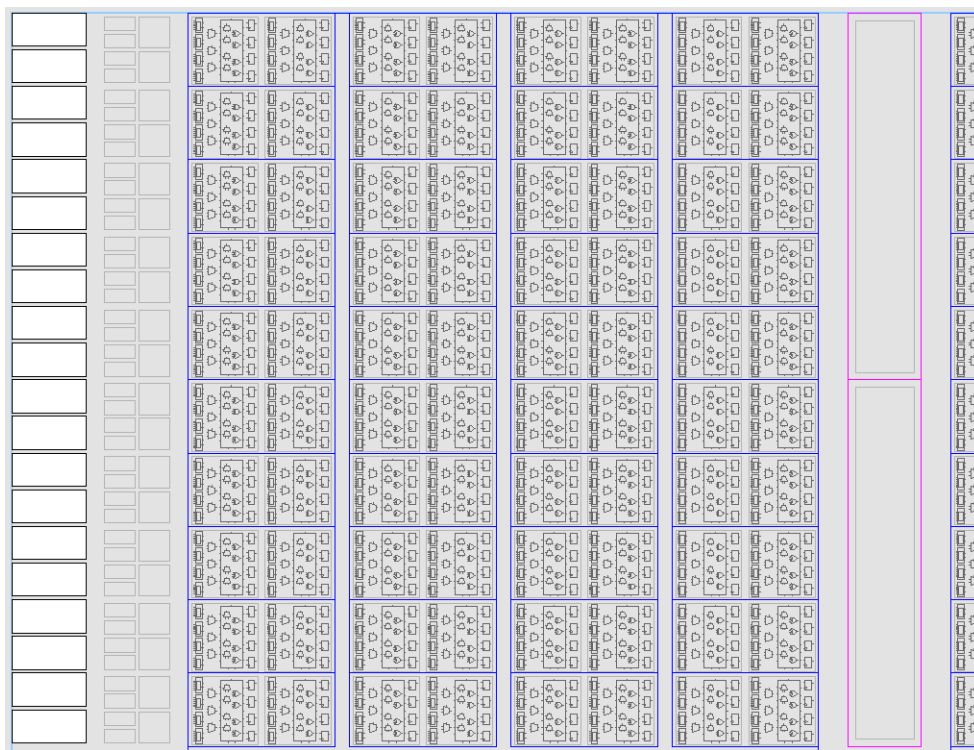


Figure 8-2: Device View Up Close

CLBs, SLICES and BELs are only visible when the zoom level is close enough to display them.

Primitive logic instances can be assigned to the appropriate sites displayed. ISE placement results can be imported to display the logic assignments.

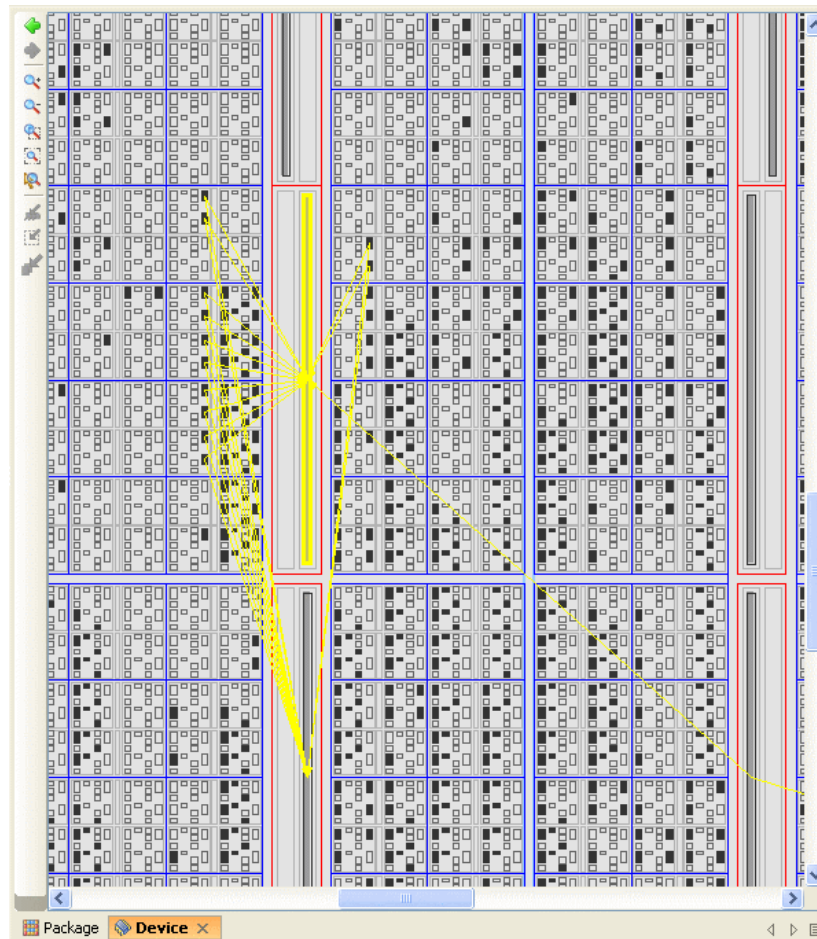


Figure 8-3: Place Instance as Rectangles in the Device View

At the zoom level shown above, placed instances appear as rectangles within a SLICE. When the zoom level increases, logic symbols are displayed.

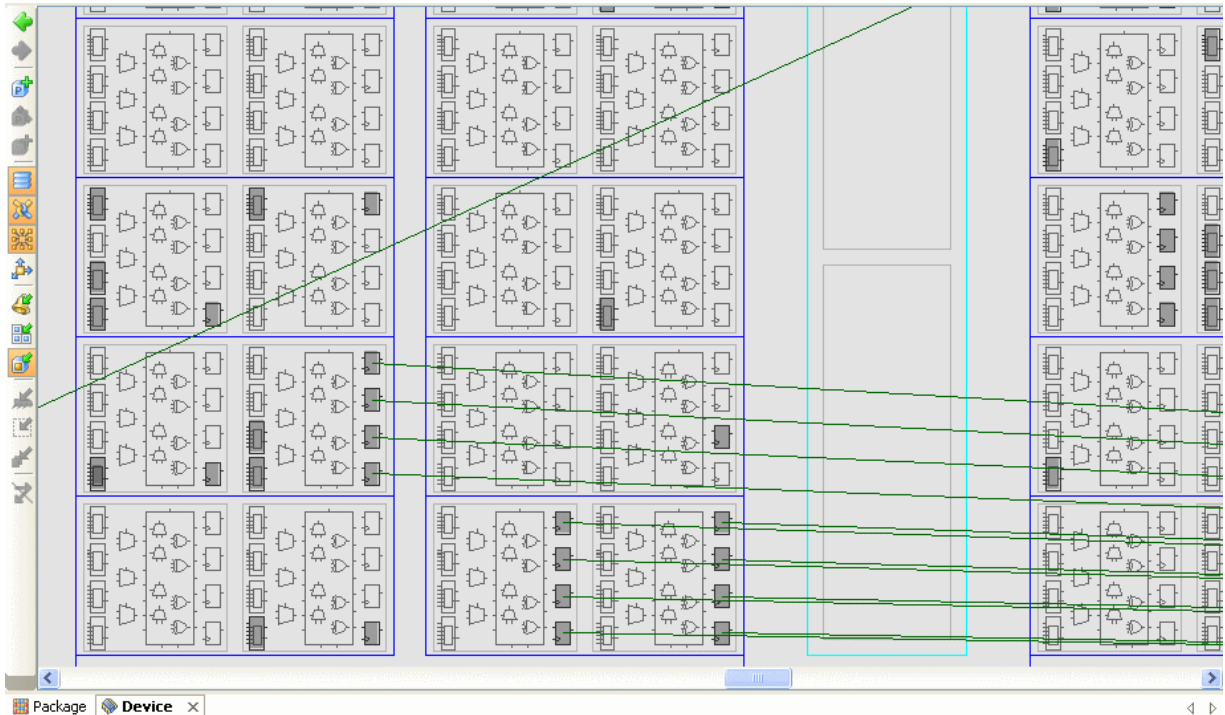


Figure 8-4: Logic Symbols in the Device View

Logic can be assigned to specific sites which will generate LOC placement constraints. They can be assigned to a SLICE or to specific gates using BEL level constraints. All logic imported from ISE is displayed as BEL level constraints. For more information about LOC placement constraints, see [“Working with Placement LOC Constraints,”](#) page 315.

Displaying Clock Regions

The clock regions are displayed as large rectangles indicating the periphery of the various device clock regions. These outlines can be used to help guide floorplanning for critical circuitry. The Clock Regions view can be used to select the various clock regions.

Select the **Window > Clock Regions** command to invoke the Clock Regions view. Clock regions can be selected and their resource statistical properties are displayed. Once implementation results have been imported, the clock placement statistics can be viewed.

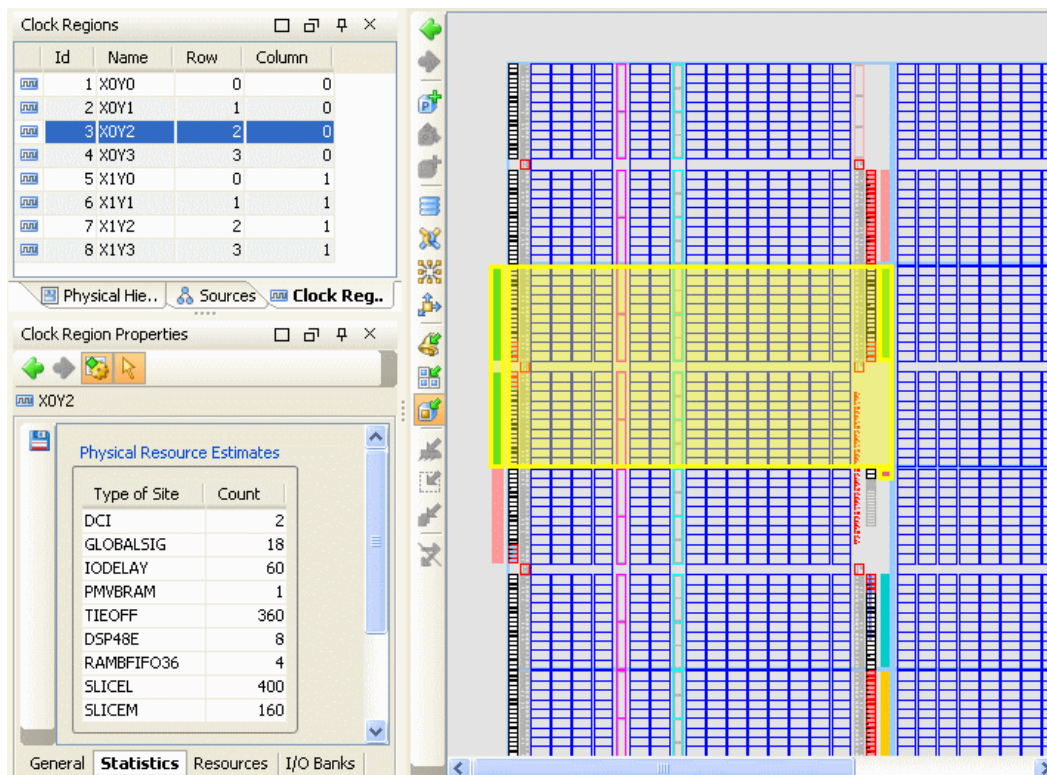


Figure 8-5: Clock Region Resource Statistics

Selecting the Clock Region will result in the associated I/O banks and clock related logic sites also being selected.

The display color for Clock Regions in the Device view can be changed using the Tools > Options > Themes > Device dialog box.

Printing the Device View

You can print the Device view using the **File > Print** command. The current viewable area is printed. To print the entire Device view, zoom to fit and then print.

Opening Multiple Device Views

Multiple Device views can be opened for the same floorplan. This allows you to work on different areas of the device. Open a second Device view display by selecting **Window > New Device View**. A separate tab will be established for Device view (2). The view can then be split using the dragging technique described in the [“Splitting the Workspace.”](#)



Figure 8-6: Displaying Multiple Device Views

Using the Schematic View

A Schematic view can be generated for any level of the logical or physical hierarchy view. The Schematic view can be used to view design interconnect, hierarchy structure or to trace signal paths for either the elaborated RTL netlist or synthesized netlist. For more information about analyzing RTL netlist, see [Chapter 6, “Creating and Analyzing the RTL Design.”](#)

Logic can be selected directly from the Schematic view for use in floorplanning in the Device view.

To create a Schematic view:

1. Select one or more logic elements.
2. Right-click and select **Schematic** from the popup menu, or select the Schematic toolbar button shown below.



Figure 8-7: Schematic Toolbar Button

The Schematic view displays the logic instances or nets selected. If only one instance was selected, the module appears with all pins displayed.

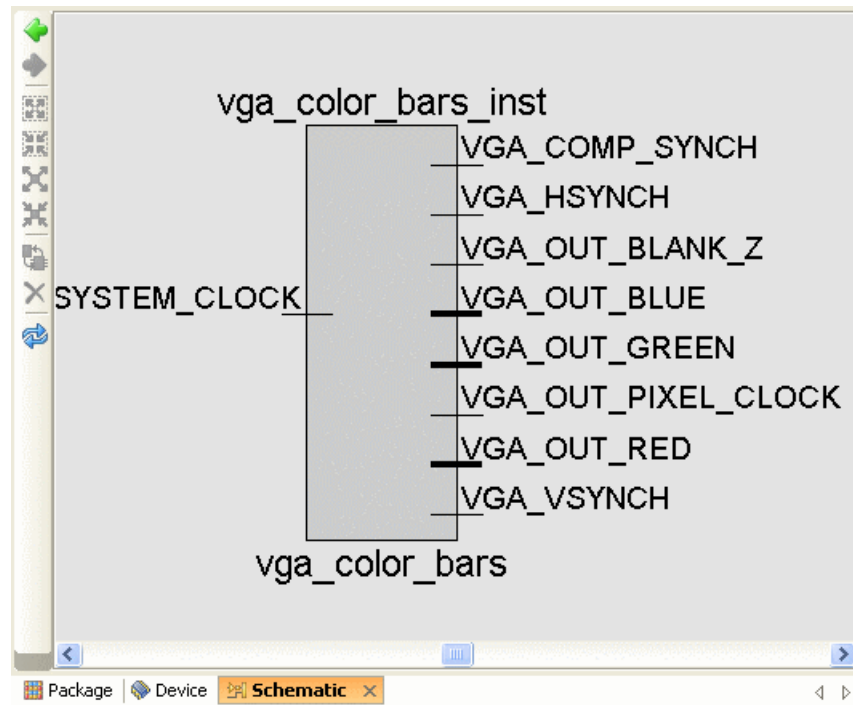


Figure 8-8: Schematic View

Selecting objects in the Schematic view will also select them in all other views. If placement results have been imported, the logic and paths are displayed in the Device view.

Viewing Logic Hierarchy in the Schematic View

All upper levels of hierarchy are displayed as concentric rectangles when a Schematic view is generated.

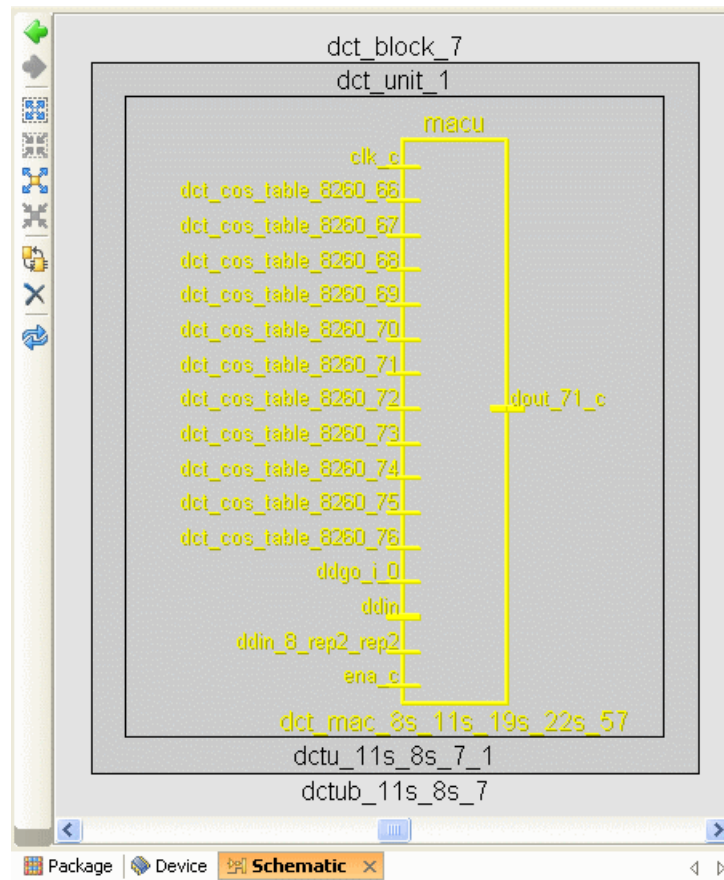


Figure 8-9: Viewing Hierarchy in the Schematic

Notice that no pins are displayed for the upper levels of hierarchy in the figure above. This makes the Schematic view more readable in most cases. Module pins and logic can be individually expanded or collapsed. Logic can be selectively expanded either from individual pins, instances or the entire logic content inside or outside the module.

To expand module pins for a selected module, use the **Toggle Autohide Pins** popup menu command or the **Toggle Autohide Pins for selected instance** toolbar button in the Schematic view.



Figure 8-10: Toggle Autohide Pins for Selected Instances Toolbar Button

Expanding Logic from Selected Pins

Several options exist to expand logic from a pin. Double-clicking on a pin will cause the logic net to expand all the way until the next primitive logic elements. Thick wires indicate busses. Busses will expand to include all bits of the bus. Expansion of signals will go beyond hierarchical boundaries, as shown below.

Other expansion options exist to expand logic all the way to the next set of Flip-Flops or all the way to the I/Os. If too much logic is selected for expansion, a dialog box may appear indicating that the logic selected is quite large and may not be suitable for schematic viewing.

To view the logic expansion options, select a pin or an instance and select the **Expand Cone** popup menu command to view the options.

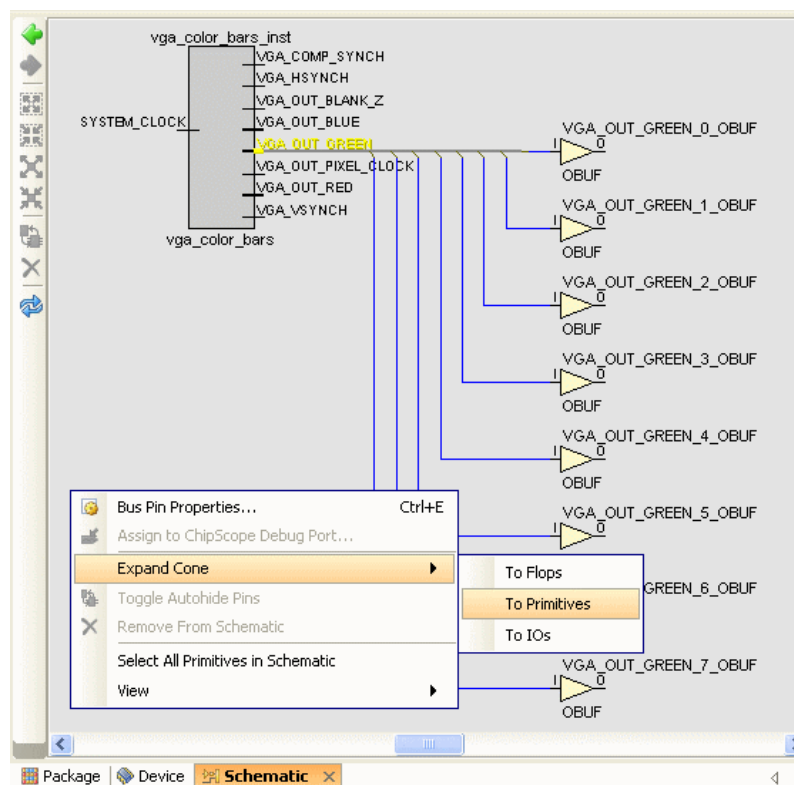


Figure 8-11: Expansion of Signals in Schematic View





The available Expand Cone logic expansion options are:

- **To Flops**—Appends the view to display the entire cone of logic to the first Flops or to any sequential element, such as Block RAMs, FIFOs, and embedded processors.
- **To Primitives**—Appends the view to display the entire cone of output logic to the first Primitives. This is also the default behavior when a pins in double-clicked.
- **To IOs**—Appends the view to display the entire cone of output logic to the IOs. This can involve a large amount of logic. PlanAhead will warn and allow you to cancel the command if more than 10 levels of logic are to be appended.

Expanding and Collapsing Logic for Selected Instances or Modules

All of the logic contained either inside of a selected module or outside in the next level of hierarchy can be expanded or collapsed instantaneously. A new set of commands can be run on either a single module or multiple selected modules. These commands are available from the right-click popup menu or from the Schematic view toolbar buttons.

Table 8-1: Schematic View Toolbar

Toolbar Button	Command	Description
	Expand all logic inside selected instance	Expands all logic inside selected instance.
	Collapse all logic inside selected instance	Collapses all logic inside selected instance.
	Expand all logic outside selected instance	Expands all logic outside selected instance.
	Collapse all logic outside selected instance	Collapses all logic outside selected instance.

The commands are intended to display all logic associated within a level of hierarchy, as shown in the example below in which the **Expand Inside** command is used.

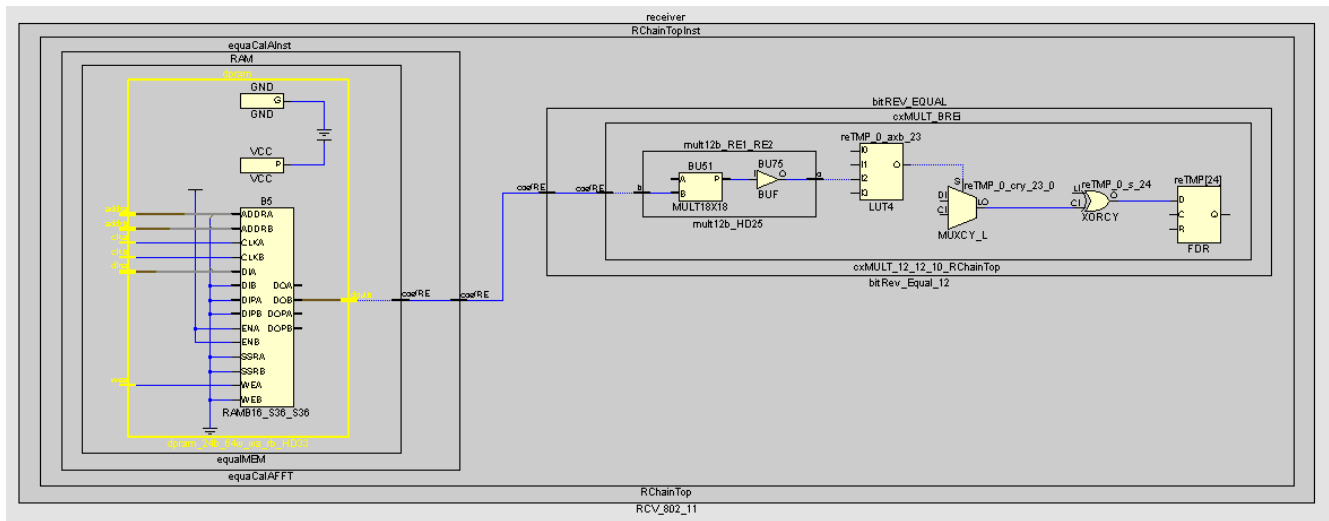


Figure 8-12: Displaying All Logic Within a Level of Hierarchy

Traversing the Schematic Hierarchy

Double-clicking on a hierarchical instance will collapse all currently displayed logic and expand the logic within the selected module. To go up a level of hierarchy, use the **Expand Outside** command in conjunction with the **Collapse Inside** command.

Regenerating a Schematic View

Occasionally, after several expand and collapse commands, the Schematic view may look a bit jumbled. You can force a Schematic view regeneration to clean up the display by selecting the Regenerate schematic toolbar command in the Schematic View.



Figure 8-13: **Regenerate Schematic Toolbar Button**

This command will redraw the active Schematic view.

Selecting Objects in the Schematic View

Objects are selected by left clicking on them. The **Ctrl** key can be used to select multiple objects. Multiple instances, ports and nets can be also selected by using the **Select Area** command and drawing a rectangle around them. As instances are selected in the Schematic view, they are also selected in all other views. This cross selection works both ways. If objects are selected or highlighted in either of these two views, they are also highlighted in the Schematic view.

The Schematic view popup menu also has some selection options which are covered later in this section of the document.

Removing Objects From the Schematic View

You can remove selected objects and their associated connectivity using the Remove Selected Elements From Schematic toolbar command in the Schematic View.



Figure 8-14: **Remove Selected Elements From Schematic Toolbar Button**

Printing the Schematic View

You can print the Schematic view using the **File > Print** command. The current viewable area is printed. To print the entire Schematic view, zoom to fit and then print.

Schematic View Specific Popup Menu Commands

Instances and nets may be selected within the Schematic view for manipulation. The common popup menu commands are covered in the [“Using Common Popup Menu Commands”](#). The Schematic view commands and a brief description of each is as follows:

- **Expand Cone**—Appends the view to display the entire cone of input logic either to the first Primitives, Flops or to the I/Os.
- **Toggle Autohide Pins**—Toggles the display of module pins for selected modules.
- **Remove Selected Elements From Schematic**—Removes the selected objects from the schematic.
- **Expand Inside**—Expands all logic contained inside of selected modules.
- **Expand Outside**—Expands all logic contained outside of selected modules. The expansion will only occur on the parent module logic.
- **Collapse Inside**—Collapses all logic contained inside of selected modules.
- **Collapse Outside**—Collapses all logic contained outside of selected modules. The collapsing will only occur on the parent module logic.
- **Select All Primitive in Schematic**—Selects all displayed primitive logic in the active schematic fit view.
- **Select Primitive Parents** (only available when instances are selected)—Selects all of the parent logic modules of the selected logic.

Annotating Design Information in the Schematic

Annotating Slack, Fanout and Values onto Schematic Pins

The PlanAhead Options dialog box Schematic settings enable you to tag source pins with *Fanout* values and destination pins with *Slack* values. Slack values are not displayed until after TimeAhead is run.

The PlanAhead Options dialog box is available by selecting **Tools > Options**, and clicking **Schematic** in the Options dialog box.

1. To annotate these values, you must first set the Attribute type field to **Pin**.
2. Select the desired values to annotate on the left side of the dialog box below, and use the arrow indicators to move them to the right side labeled Displayed DB Attributes.
3. Click **OK**.

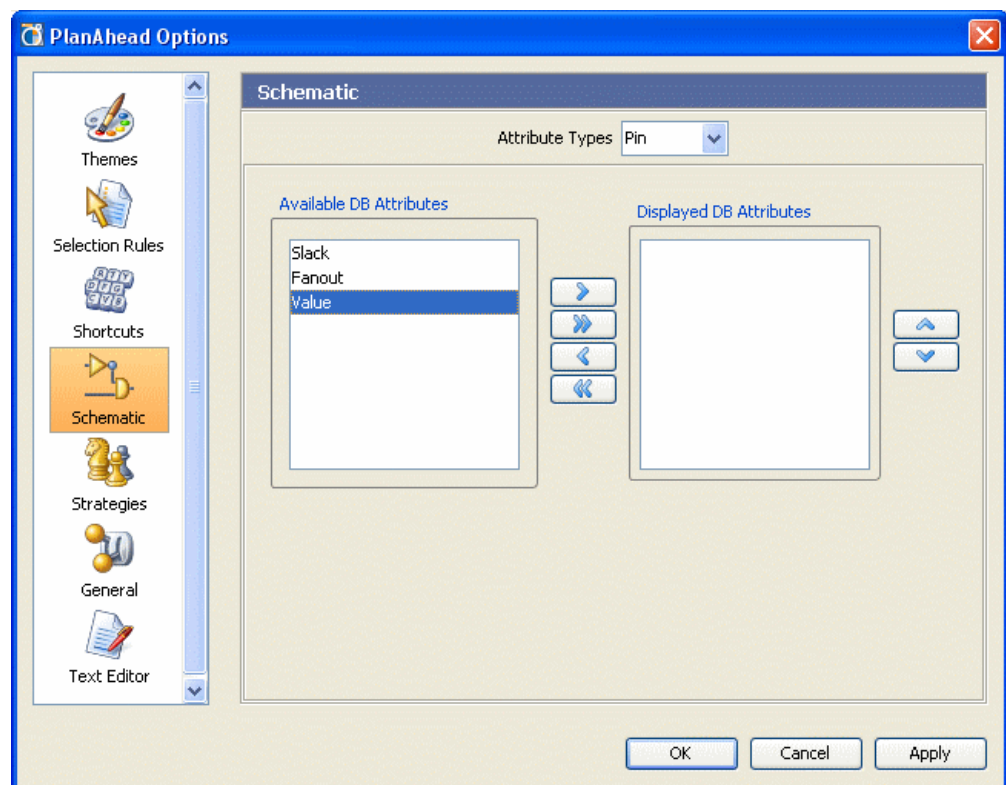


Figure 8-15: PlanAhead Options: Schematic Pin Annotation

An example of the resulting pin annotation is shown below.

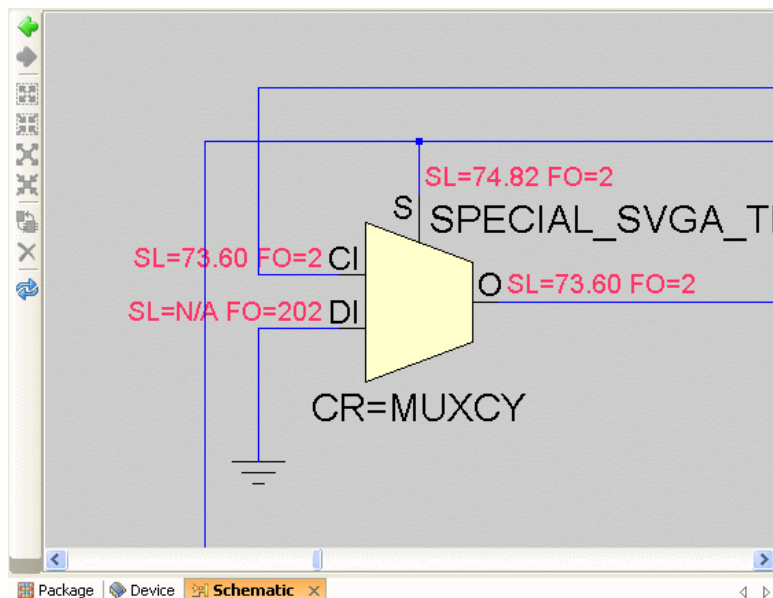


Figure 8-16: Annotated Pins in the Schematic View

Annotating Cell References and Instance Equations onto Instances

The PlanAhead Options dialog box Schematic settings enable you to tag instances with *Cell References* and *Instance Equation* values.

The PlanAhead Options dialog box is available by selecting **Tools > Options**, and clicking **Schematic** in the Options dialog box.

1. To annotate these values, you must first set the Attribute type field to **Instance**.
2. Select the desired values to annotate on the left side of the dialog box below and use the arrow indicators to move them to the right side labeled Displayed DB Attributes.
3. Click **OK**.

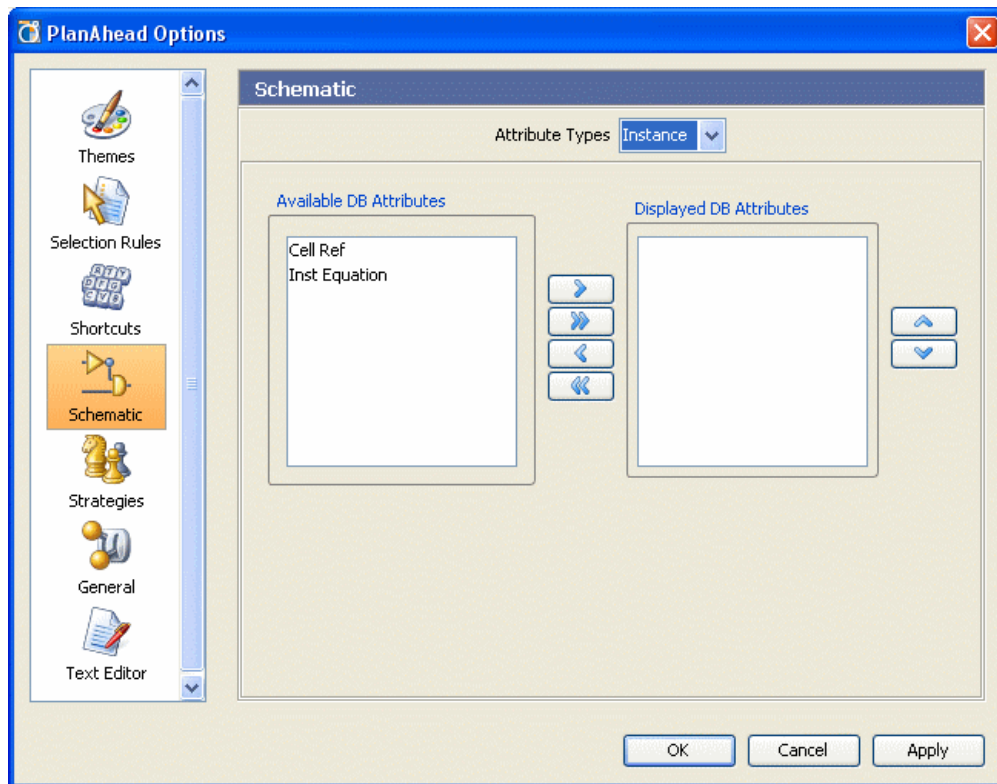


Figure 8-17: PlanAhead Options: Schematic Instance Annotation

An example of the resulting instance annotation is shown below.

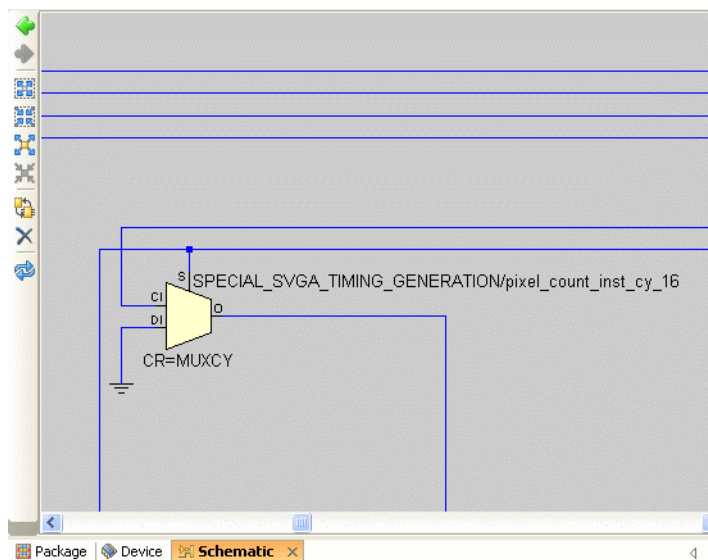


Figure 8-18: Annotated Instances in the Schematic View

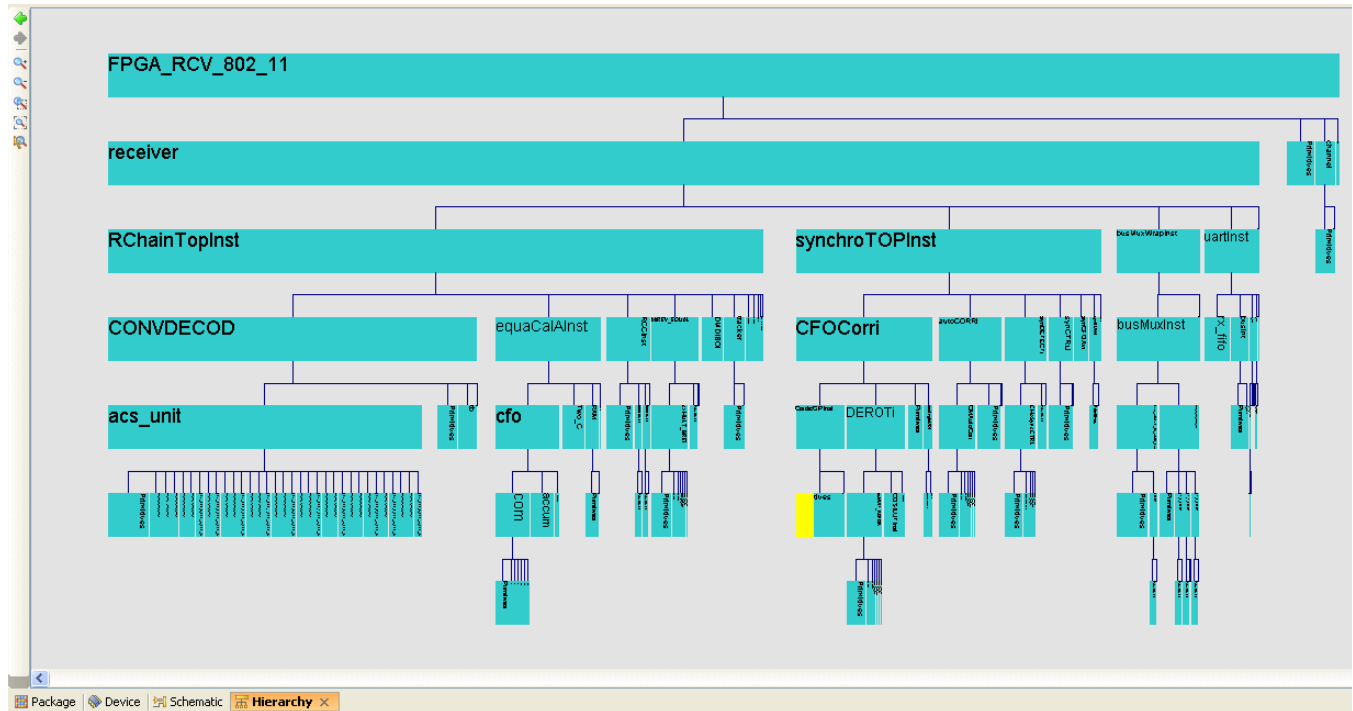


Figure 8-20: Hierarchy View

Only hierarchical instances are displayed in the Hierarchy view. Primitive logic is grouped into folders that are represented as submodules. Refer to the “Using the Netlist View” for more information about Primitive logic folders. The widths of the blocks in the Hierarchy View are based on the relative FPGA resources, including LUTs, flops, block RAMs, and DSP48s.

Selected logic is highlighted so you can quickly see where critical logic resides in the design. When logic contained in a module is selected, the module highlights proportionally to the amount of logic selected.

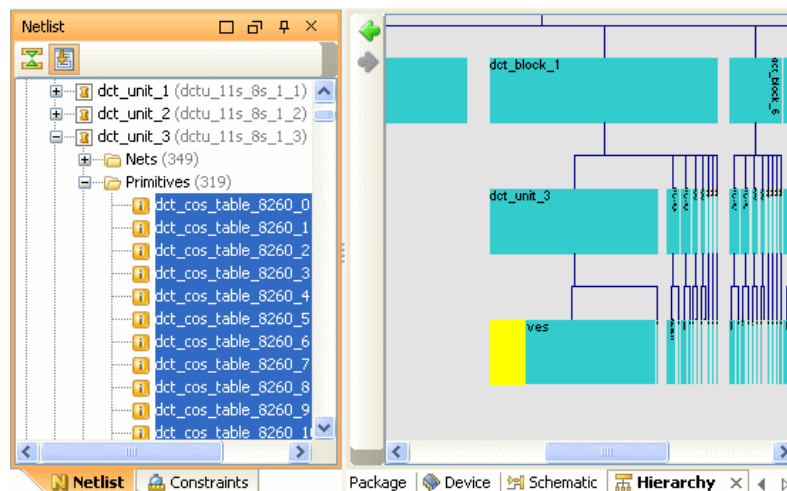


Figure 8-21: Display Percentage of Logic Selected in Module

A sub-hierarchy for any submodule can be displayed by double-clicking on the module in the Hierarchy view.

To select logic parent modules for Pblock assignment in this view, use the **Select Primitive Parents** command.

Using the Properties View

All PlanAhead objects have associated properties. As objects are selected, their properties will automatically appear in the Properties view. The Properties view has object-specific tabs that display various types of information.

The Properties dialog box is dynamic by default. As new items are selected, the view updates automatically.

Many of the object types have multiple types of properties displayed. View tabs are added to the bottom of the Properties view to accommodate various types of information. Select the different tabs to display or modify information about the selected object.

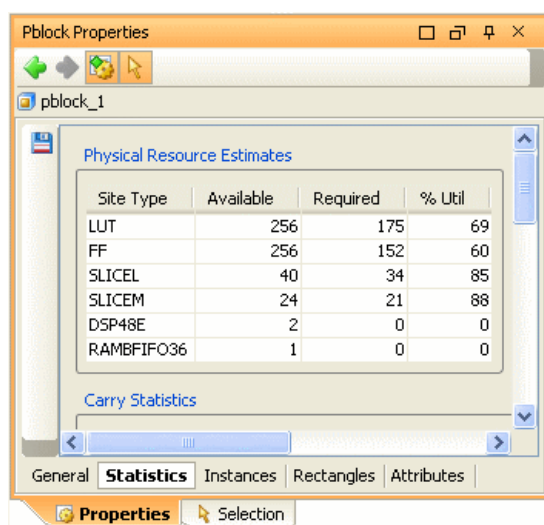


Figure 8-22: Property View With Tabs

Using the Properties View Toolbar

The Properties view toolbar contains the following commands:

Table 8-2: Properties View Toolbar








Toolbar Button	Command	Description
	Previous object	Reverts to the previously selected objects.
	Next object	Reverts to the next selected objects (This key is only enabled after a Previous object command).
	Automatically update the contents of this window for selected objects	Toggles the Properties view to auto update as new objects are selected or remain static on the originally selected object.

Table 8-2: Properties View Toolbar

Toolbar Button	Command	Description
	New	Adds a new object. This option is only available for certain object types and in specific view panes
	Delete	Deletes an object from within one of the property tabs.
	Export statistics	Saves data to file for later analysis. Available from the Statistics tab for the Pblock, Clock Region and Instance Properties View only.
	Select/Unselect object	Sometimes the object whose properties you are viewing may become unselected. Use this button to select/unselect this object.

Using the Netlist View

The netlist is a hierarchical representation of the logic design beginning with the top-level netlist name followed by top-level modules. The Netlist view displays the logic instances and nets contained in the design. The netlist can be navigated by expanding and collapsing the logic tree. Scroll bars are used to view the entire netlist tree.

The netlist tree dynamically expands and scrolls to display netlist objects whenever they are selected in other views. This default is considered the most useful. To disable this feature, select the **Automatically scroll to selected objects** toolbar button in the Netlist view.

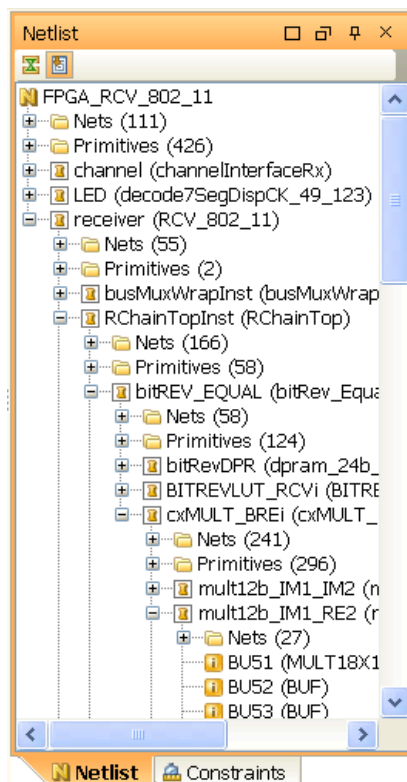


Figure 8-23: Netlist View

Collapsing the Netlist Tree

The entire netlist tree can be collapsed by selecting the **Collapse All** toolbar button in the Netlist view. For more information, see [“Using the View Specific Toolbar Commands.”](#)

The Netlist tree collapses to display only the top-level logic modules.

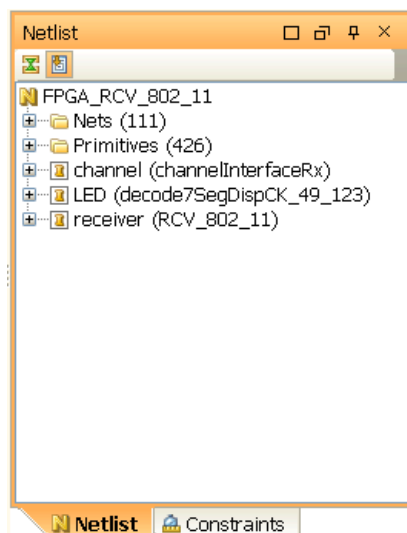


Figure 8-24: Collapsed Netlist Tree

Using the Primitives Folder

When a module contains primitive logic, the primitive logic is placed in a *Primitives* folder. This helps condense the display the modules in the Netlist view.

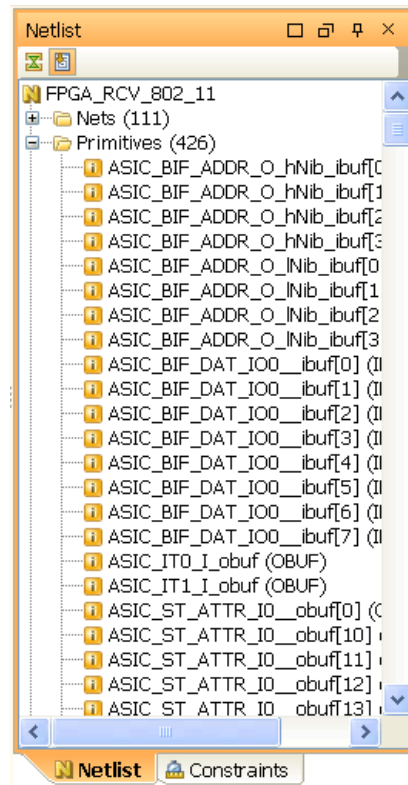


Figure 8-25: Primitives Folder in the Netlist View

You can assign the Primitives folder directly to a Pblock resulting in all primitives to be assigned.

Note: Netlist updates may require reassignment of the Primitives folder to the Pblock as logic names may have changed during re-synthesis.

Using the Nets Folder

The Nets folder contains all of the nets and busses defined at any level of hierarchy. Busses can be expanded to show each individual bit.



Figure 8-26: Nets Folder in Netlist View

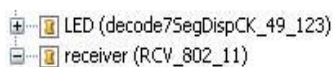
As nets are selected, they are highlighted in the Device view. Selecting a bus will highlight all nets contained in it. Nets can be viewed in the Schematic view.

Understanding the Netlist View Icons

Various icons are used to represent the state of the netlist logic.

Hierarchical netlist modules

Hierarchical netlist modules or “instances” are displayed with a yellow **I** icons, as shown below.



Hierarchical netlist modules assigned to Pblocks

Hierarchical netlist modules or “instances” assigned to Pblocks are displayed with blue checkmark icons, shown below.



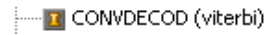
Modules derived from NGC

Modules that were derived from NGC or NGO format core netlist input files are displayed with a red square in the lower right corner, as shown below. The NGC file name is displayed in parenthesis.



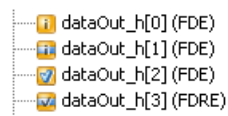
Black box Modules

Modules that do not have netlists associated with them are displayed with a yellow **I** icon and a dark background. This could be a result of a search path not being specified during Project creation or missing portions of the design.



Primitive logic instances

Primitive logic instances *without* placement constraints assigned display as an “i” inside a yellow rectangle. Primitive logic instances *with* placement constraints assigned display a yellow rectangle with a blue stripe. Primitive logic instances assigned to a Pblock display a blue checkmark inside a yellow rectangle. Primitive logic instances that are placed and assigned to a Pblock display a checkmark and blue stripe in a yellow rectangle. Notice the logic types are also displayed.



Selecting Logic in the Netlist View

Instances can be selected and commands applied to them using the menu, Toolbar or mouse context sensitive menu (popup menu).

The **Shift** key or the **Ctrl** key can be used to select multiple elements in the Netlist view for use with most commands. Selected logic is highlighted in the Netlist view.

Logic selected by any means in PlanAhead is also displayed as selected in the Netlist view. The Netlist tree will automatically expand to display all selected logic. You may need to scroll the tree to view all selected logic. Collapsing the Netlist tree does not unselect logic.

Using the Netlist View Specific Popup Menu Commands

For more information on the popup menu commands available from the Netlist View, see [“Using Common Popup Menu Commands.”](#)

Using the Constraints View

PlanAhead provides a view of all of the timing constraints defined in the floorplan. constraints are floorplan-specific, and can vary between floorplans in the same project. You can experiment with different constraints, devices, IO pins, etc.

The Constraints view interface also allows you to modify defined values and to create new constraints. To view the timing constraints defined in the design, select the Constraints view tab, or select **Window > Constraints**.

The Constraints view will appear.

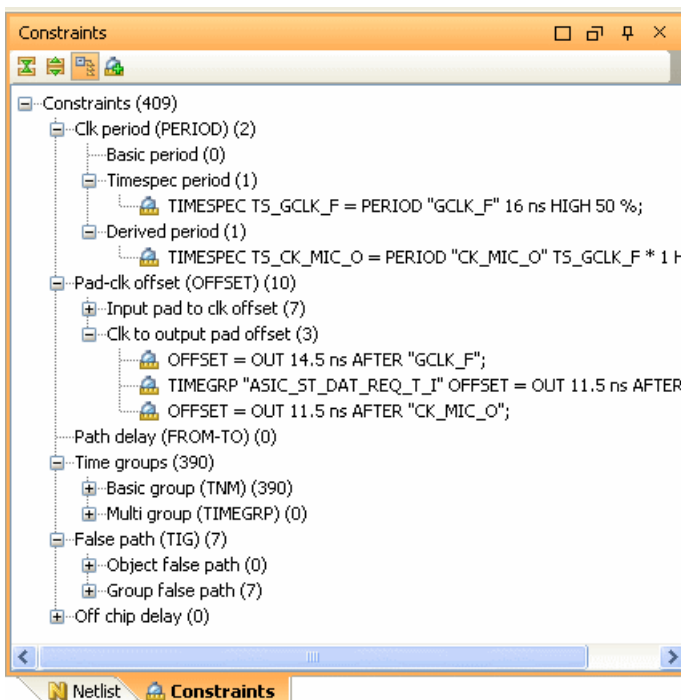


Figure 8-27: Constraints Grouped by Type

The constraints are displayed in two different ways. As shown above, categorically sorted by type allowing expansion and collapsing of the levels of constraint types. Notice the number of each type of constraint is displayed in parenthesis.

To view a list of all timing constraints, click the **Group by type** toolbar button in the Constraints view.



Figure 8-28: Group by Type Toolbar Toggle Button

When timing constraints are displayed as a list, they look as follows:

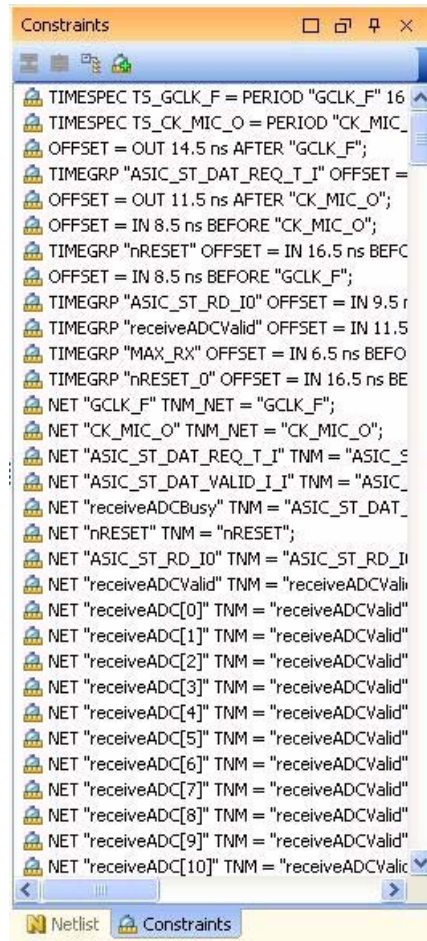


Figure 8-29: Timing Constraints Grouped by List

Modifying Timing Constraints Values

Most constraint values can be modified by selecting a constraint and viewing the Constraints Properties. The appropriate changeable values for the constraint are shown in the dialog box.

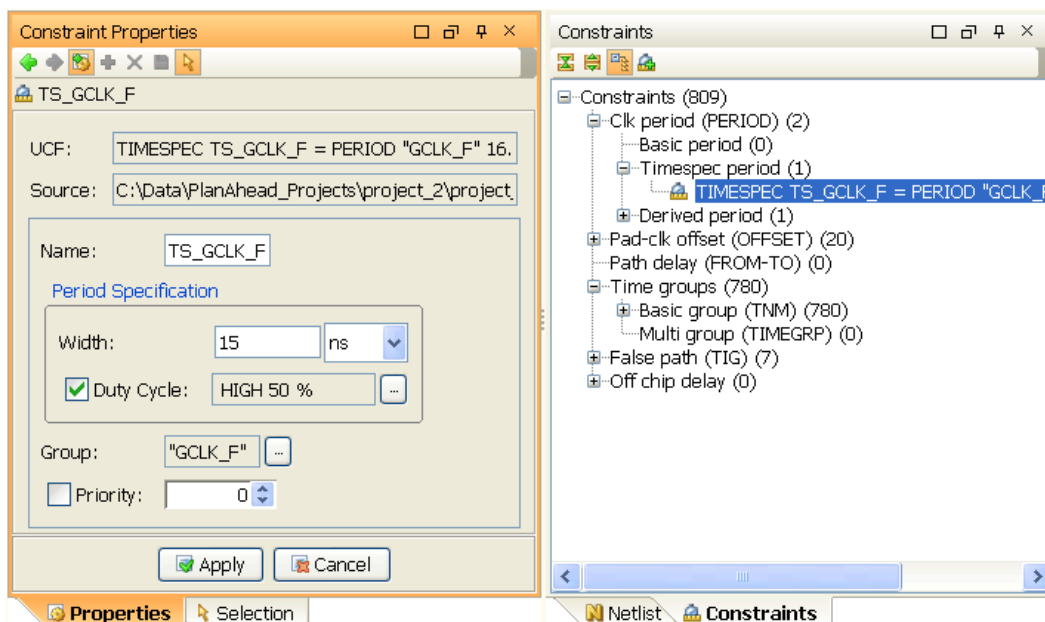


Figure 8-30: Modifying Timing Constraints Properties

The various dialog boxes for each constraint type are too numerous to describe here. Use the correct syntax when defining constraints values. Refer to the Xilinx® *Constraints Guide* for more information on constraints and constraints syntax. Once changes have been made, click **Apply** to accept the changes, or click **Cancel** to deny the changes.

Note: Failure to click the Apply button will not initiate any changes to constraints values.

Adding New Timing Constraints

To add new timing constraints:

1. Click the **Create new constraint** toolbar button in the Constraints view.



Figure 8-31: Create New Constraint Toolbar Button

The New Timing Constraint dialog box will appear.

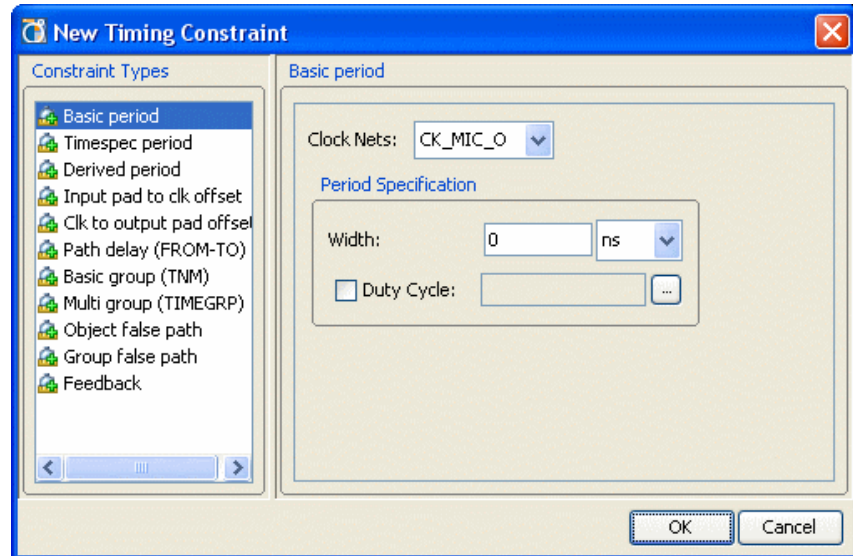


Figure 8-32: Create New Timing Constraint Dialog Box

2. Select the type of constraint you wish to create on the left.
The appropriate fields will be displayed on the right.
3. Define the constraint values using the correct syntax. Refer to the *Xilinx Constraints Guide* for more information on constraints and constraint formats.
4. To accept the changes, click **OK**.

Removing Timing Constraints

To remove the constraint from the Floorplan, select a constraint or group of constraints in the Constraints view, and select **Delete** from the popup menu. You will be prompted to remove the selected constraint(s) prior to the removal of the constraint.

Note: Due to the interdependence between timing constraints, removing one constraint may result in the removal of several other related constraints.

Note: Adding, editing, and deleting timing constraints cannot be undone.

Using the Physical Hierarchy View

The Physical Hierarchy view displays the hierarchical partitioning structure of the design. Physical design navigation is performed by expanding and collapsing the tree widget elements. Tree elements can be selected and commands applied to them using the menu, toolbar or context sensitive menu (mouse popup menu). Press **Ctrl** or **Shift** keys to select multiple elements for use with some commands.

The physical hierarchy tree is dynamic and will automatically expand and change when physical hierarchy is manipulated. As objects are selected in other views, the appropriate elements are highlighted in the Physical Hierarchy view.

The objects displayed in the Physical Hierarchy view are Relatively Placed Macros (RPMs) and Physical Block (Pblocks). These objects may be selected in the Physical Hierarchy view for manipulation in other views.

Using the ROOT Design Pblock

The physical hierarchy begins with the Floorplan name followed by the top-level design called “ROOT”. ROOT is considered as the top-level Pblock within PlanAhead. As lower level Pblocks are created they are displayed in a hierarchical fashion under the Pblock Folder, with Child Pblocks appearing under their parent Pblock.

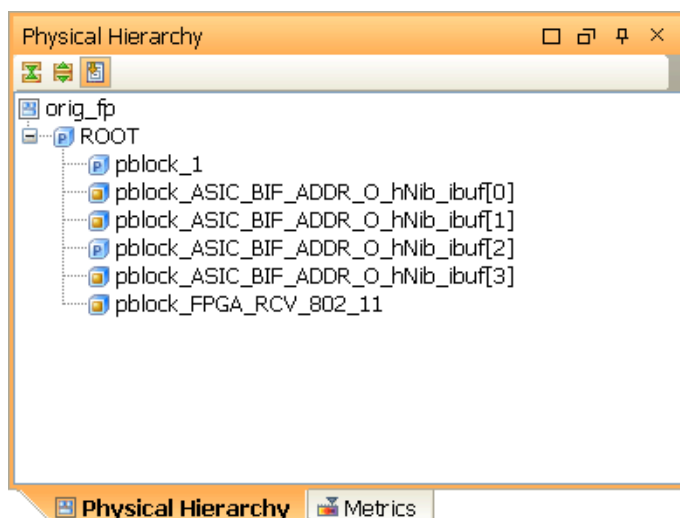


Figure 8-33: Physical Hierarchy View

Selecting a Pblock will also select all of the logic assigned to it.

Understanding the Physical Hierarchy Icons

The Physical Hierarchy tree utilizes several icons that can help you identify the state of the various objects. This display automatically updates as the physical hierarchy is manipulated.

As Pblocks are created, they appear in a hierarchical fashion in the Physical Hierarchy view. Each folder type in the Physical Hierarchy view displays a number in parenthesis detailing the number of objects in that folder, as shown in Figure 8-34.

Each instantiation of an RPM is displayed in the Physical Hierarchy View. If RPMs are assigned to Pblocks, they appear in an RPM folder under the Pblock. Selecting an RPM in the Physical Hierarchy view will also select all of the logic contained in the RPM.

Pblocks with instances assigned

Pblocks with instances assigned and *with* rectangles defined in the Device view appear as blue three-dimensional cubes with a yellow center.



Pblocks with instances assigned and *with no* rectangles defined in the Device view appear as blue two-dimensional squares with a yellow center.



Pblocks with no instances assigned

Pblocks with no instances and *with* rectangles defined assigned appear as blue three-dimensional cubes with a blue P in the center.



Pblocks with no instances assigned with *no* rectangles defined appear as blue two-dimensional squares with a blue P in the center.



Working with Relatively Placed Macro (RPMs)

Relatively Placed Macros (RPMs) that exist in the design are listed under the RPMs folders. RPMs can be assigned to Pblocks. If RPMs are assigned to Pblocks, they appear in an RPM folder under the Pblock. Each instantiation of an RPM is displayed in the Physical Hierarchy View.

Selecting an RPM in the Physical Hierarchy view will also select all of the logic found in the RPM.

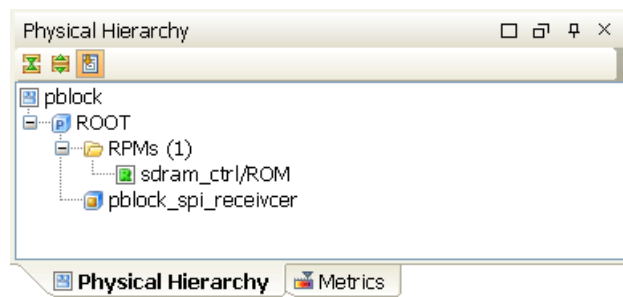


Figure 8-34: RPM Folders Displayed in Physical Hierarchy

The RPM Properties view displays all instances found in the RPM.

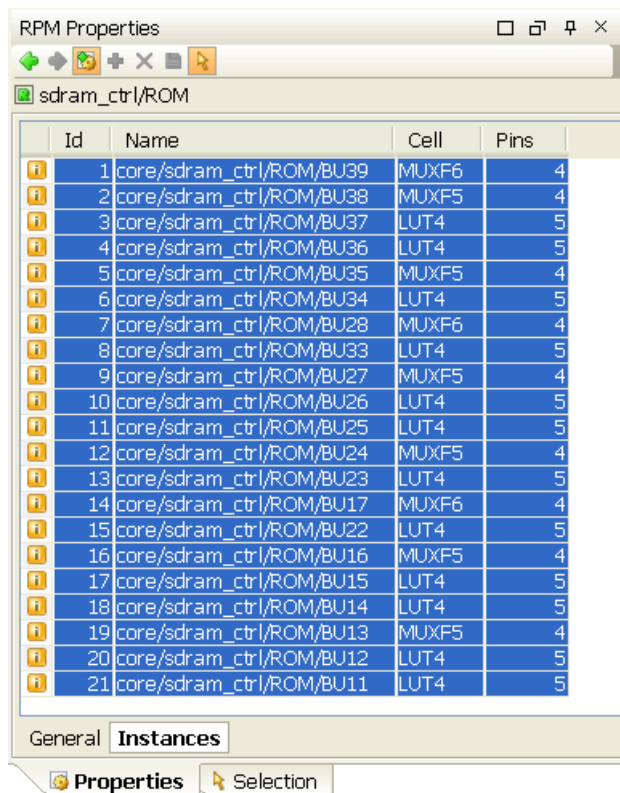


Figure 8-35: Viewing All Instances in the RPM

When you assign RPM logic to a Pblock, the RPM size information will be displayed in the Pblock Statistics Properties.

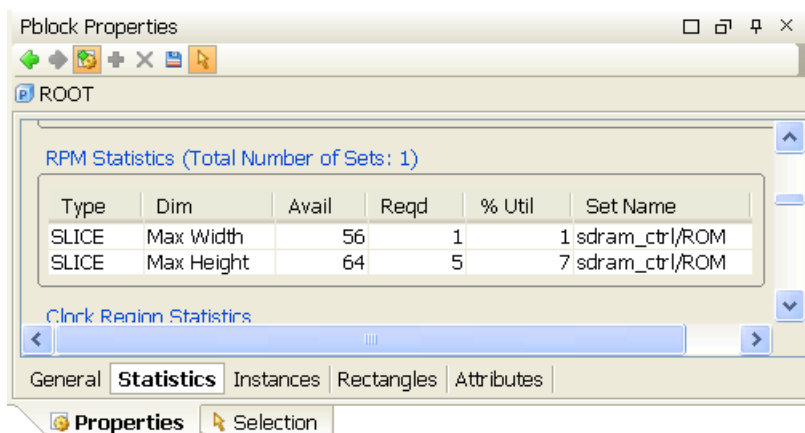


Figure 8-36: Pblock Properties: RPM Statistics

The number of RPMs assigned to the Pblock along with the tallest and the widest RPMs are displayed. This RPM size information is helpful when shaping the Pblock as the overall height and width utilization percentage is displayed.

To save this data, click the Save icon of the Pblock Properties dialog box, and specify a file name for data.



Figure 8-37: **Save Pblock Statistics to File**

It is recommended that you increase the size of your Pblocks if they contain a large number of RPMs. This enables the implementation tools to fit the RPMs into the area.

Using Common Popup Menu Commands

Many of the popup menu commands are available from multiple views. The commands listed below are common across many of the PlanAhead views and environments.

Some of the following commands are only available when logic is pre-selected. The popup menu commands vary depending on which view is active and what object type is selected.

The common right-click commands and a brief description of each is as follows:

- **Instance Properties / Pblock Properties / Net Properties**—Displays the appropriate “Properties” dialog box.
- **Delete**—Deletes the selected objects after a confirm dialog box is displayed.
- **Unplace**—Removes the selected placement constraints.
- **Assign**—Assigns the selected instance to an existing Pblock. A Pblock selection dialog box enables you to select which Pblock to assign the selected instances.
- **Unassign**—Removes the selected instances assignments from a Pblock.
- **Draw Pblock**—Enables you to draw a rectangle in the Device view. The Pblock is created with the selected instances assigned.
- **New Pblock**—Creates a new Pblock in the Physical Hierarchy view with the selected instances assigned. No rectangle is created.
- **Clear Rectangle**—Removes the selected rectangles from the Pblock. The Pblock is not deleted.
- **Set Pblock Size**—Enables you to draw a new rectangle in the desired location. All other existing rectangles are removed.
- **Add Pblock Rectangle**—Enables you to draw an additional rectangle for the Pblock. This is helpful when trying to create non-rectangular shapes for Pblocks.
- **Select Children**—Selects all child Pblocks for all selected Pblocks.
- **Select Primitives**—Selects all primitive logic objects in the current Schematic view. If modules are selected, only the primitive logic objects inside of those modules will be selected. This command is available in other views as well.
- **Select Primitives Parents**—Selects all parent modules of the selected primitive logic objects. If modules are selected, they remain selected. The command does not ascend the hierarchy and selects a module’s parent. This command is available in other views as well.
- **Highlight Primitives**—The primitive logic that belongs to the selected modules is highlighted using the color selected in the secondary popup menu. This command is available in many views.

If a group of modules is selected, you can highlight each module primitive using a unique color by selecting the **Cycle Colors** option in the secondary popup menu. This command is very helpful when displaying placement locations for groups of logic

hierarchy. The module icons in the Netlist view are also highlighted with the same unique colors to easily identify associated objects from view to view.

- **Unhighlight Primitives**—Removes highlighting from the selected primitives. If modules are selected, the primitive logic objects inside of those modules will be unhighlighted.
- **Schematic**—Creates a new Schematic view containing the selected logic.
- **Show Connectivity**—Selects all of the nets that connect to the selected instances. This command can be run successively to sequentially select the instances that all of the displayed nets connect to. Running it again will then highlight the next group of expanded nets in the logic cone, and so on.
- **Show Hierarchy**—Invokes the Hierarchy view and displays the entire design with the modules containing the selected objects highlighted.
- **Fix Instances**—Enables you to lock placement of the logic. When logic is “fixed”, it is considered user assigned, and is exported to ISE by default.
- **Unfix Instances**—Enables you to remove locked placement.
- **Highlight**—Highlights the selected objects using the active highlight color.
- **Highlight with**—Highlights the selected objects using the selected highlight color.
- **Mark**—Places a mark symbol on all selected elements in the Device view.
- **Select**—Opens a submenu which lists all selectable objects for the location where the popup menu is invoked. This is helpful when trying to select a particular object among a set of overlapping objects.
- **View**—Opens a submenu of commands.
 - ♦ **Zoom**—Opens a submenu of commands to expand or shrink the current view or selected object.
 - ♦ **Fit Selection**—Fits the display to include all selected objects.
 - ♦ **Fit Highlight**—Fits the display to include all highlighted objects.
 - ♦ **Fit Markers**—Fits the display to include all marked objects.
 - ♦ **Options**—Invokes the PlanAhead Options dialog box (which is also accessed using Tools > Options.) For more information, see [“Customizing PlanAhead Display Options.”](#)
 - ♦ **Refresh**—Redraws and refreshes the display.
- **Metric**—Opens a submenu which lists all available metrics to display. For more information on Metrics, see [“Analyzing the I/O Pinout and Clock Logic.”](#)

Analyzing the I/O Pinout and Clock Logic

Refer to [Chapter 5, “I/O Pin Planning”](#) for more information.

Analyzing the RTL Design

Refer to [Chapter 6, “Creating and Analyzing the RTL Design”](#) for more information.

Analyzing the Synthesized Design

Reporting Design Resource and Device Utilization Statistics

Reporting Device Resource Utilization Statistics

PlanAhead provides estimates of the number of device resources utilized by the design. These device utilization statistics can be displayed or output as follows:

1. Select either the ROOT design or any Pblock in the Physical Hierarchy view.

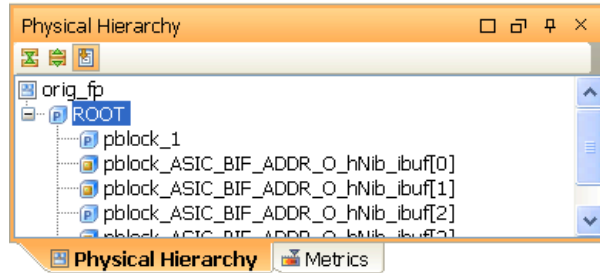


Figure 8-38: Physical Hierarchy View with ROOT Selected

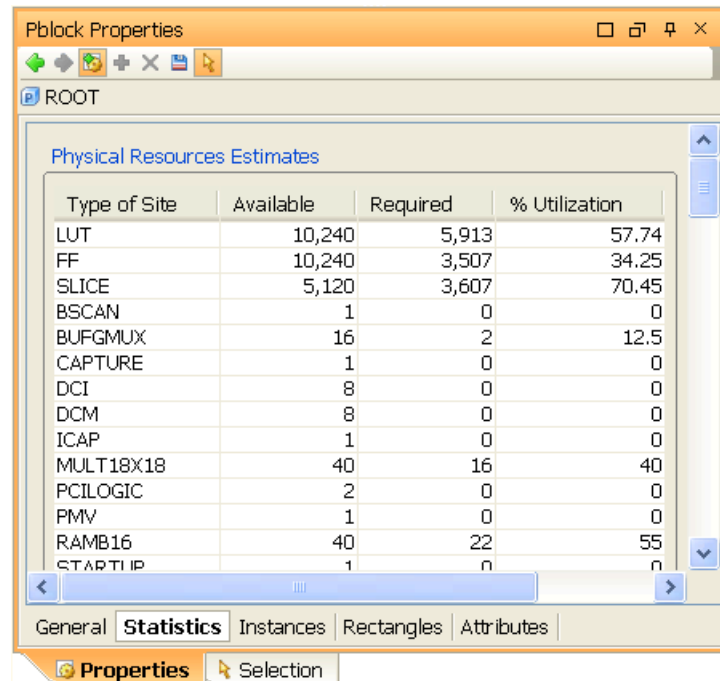
The Pblock Properties should appear in the Properties view.

2. If the Pblock Properties are not displayed, right-click on the ROOT or Pblock, and select **Pblock Properties** from the popup menu.

The Pblock Properties dialog box contains five tabs, four of which are discussed in this section.

3. Click the **Statistics** tab.

The Statistics tab displays valuable design information including overall device utilization for the various device resources, carry chain count and max length, RPM count and maximum sizes, clock names and clocked instance count, I/O utilization, and signal and primitive instance counts.



The image shows the 'Pblock Properties' dialog box with the 'Statistics' tab selected. The 'Physical Resources Estimates' section displays a table with the following data:

Type of Site	Available	Required	% Utilization
LUT	10,240	5,913	57.74
FF	10,240	3,507	34.25
SLICE	5,120	3,607	70.45
BSCAN	1	0	0
BUFGMUX	16	2	12.5
CAPTURE	1	0	0
DCI	8	0	0
DCM	8	0	0
ICAP	1	0	0
MULT18X18	40	16	40
PCILOGIC	2	0	0
PMV	1	0	0
RAMB16	40	22	55
STARTUP	1	0	0

The dialog box also includes tabs for 'General', 'Statistics', 'Instances', 'Rectangles', and 'Attributes'. At the bottom, there are buttons for 'Properties' and 'Selection'.

Figure 8-39: Pblock Properties: Statistics Tab

Exporting Estimated Hierarchical Device Utilization Statistics

You can save the displayed data to an spreadsheet file. PlanAhead enables a hierarchical style report to be generated. You can define how many levels of hierarchy to report with estimates listed for each module at each level.

1. Select the Export Statistics button to export the data to a spreadsheet file.



Figure 8-40: Export Statistics Toolbar Button

The Export Pblock Statistics dialog box appears.

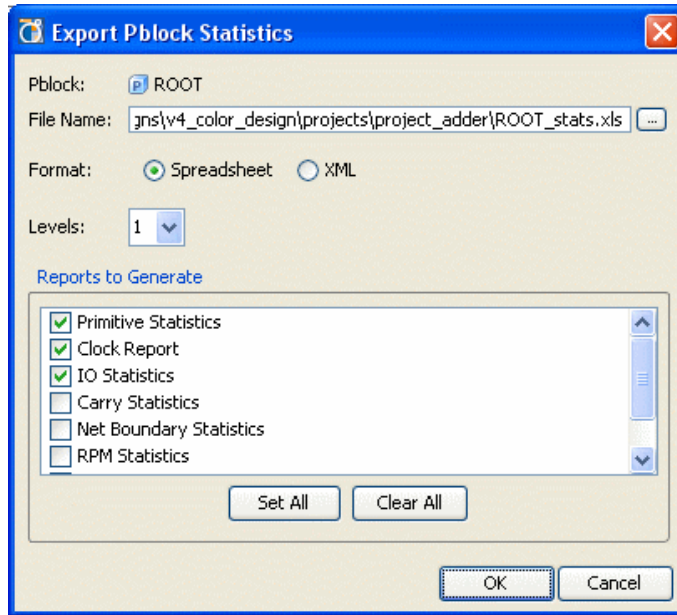


Figure 8-41: Export Design Utilization Statistics

The Export Pblock Statistics dialog box contains the following editable options:

- ◆ **File Name** - Enter the name and location of the spreadsheet file to be created.
- ◆ **Format** - Select either an XML or Microsoft Excel output file format.
- ◆ **Levels** - Indicate the number of levels of hierarchy to traverse and include in the report as separated modules.
- ◆ **Reports to Generate** - Define the types of information from the Pblock Property Statistics view to include in the output report file.

2. Select the desired options for the exported file.
3. Click **OK**.

Reporting Design Resource Statistics

PlanAhead provides estimates of the number of device resources contained in the design.

This same type of estimated statistical information can be displayed and exported from an elaborated RTL design project. Refer to the [“Elaborating and Analyzing the RTL Design.”](#)

These design resource statistics are displayed as follows:

1. Select either the top-level module or any instance module in the Netlist view.

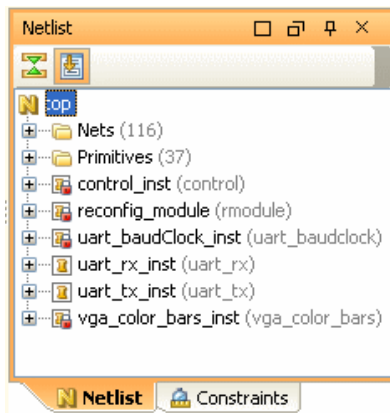


Figure 8-42: Netlist View with Top Module Selected

The Netlist or Instance properties should appear in the Properties view.

2. If the Netlist or Instance Properties are not displayed, right-click on the module, and select **Netlist Properties** or **Instance Properties** from the popup menu.

The Netlist Properties view contains one tab and should display the statistics by default. The Instance Properties dialog box contains five tabs.

3. If viewing the Instance Properties, click the **Statistics** tab.

The Statistics tab displays valuable design information including primitive instance counts, interface signal counts, clock names and clocked instance count, and carry chain count and max length.

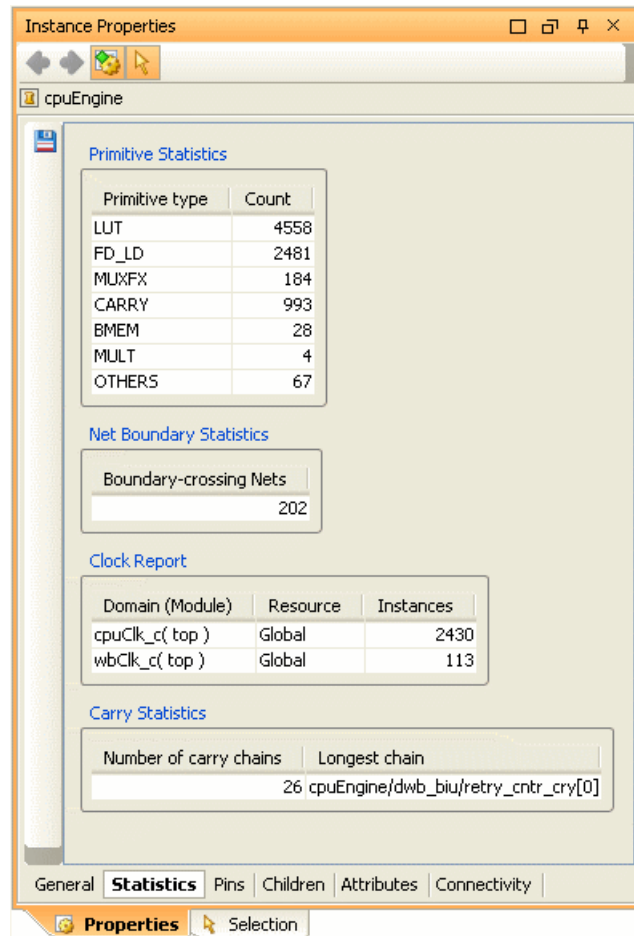


Figure 8-43: Netlist Resource Statistics

Exporting Hierarchical Design Resource Statistics

You can save the displayed data to a spreadsheet file. PlanAhead enables a hierarchical style report to be generated. You can define how many levels of hierarchy to report with estimates listed for each module at each level.

1. Select the Export Statistics button to export the data to a spreadsheet file.



Figure 8-44: Export Statistics Toolbar Button

The Export Netlist Statistics dialog box appears.

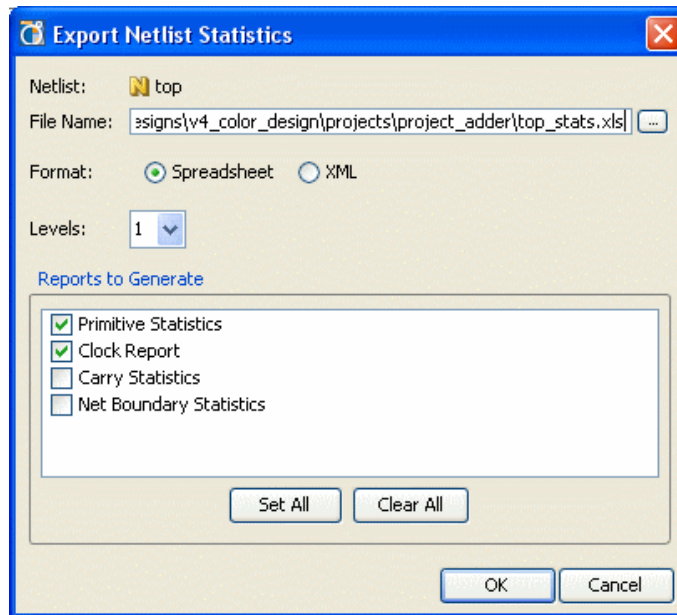


Figure 8-45: Export Netlist Resource Statistics

The Export Netlist Statistics dialog box contains the following editable options:

File Name - Enter the name and location of the spreadsheet file to be created.

Format - Select either an XML or Microsoft Excel format output file format.

Levels - Indicate the number of levels of hierarchy to traverse and include in the report as separated modules.

Reports to Generate - Define the types of information from the Pblock Property Statistics view to include in the output report file.

2. Select the desired options for the exported file.
3. Click **OK**.

Exploring the Logical Hierarchy

Refer to the [“Using the Netlist View”](#) and [“Using the Instance Hierarchy View”](#) for more information.

Analyzing the Hierarchical Connectivity

PlanAhead provides a wide variety of capabilities to examine the logic hierarchy including the ability to visualize connectivity between the various logic modules. Sometimes, it is helpful to initially create a Floorplan consisting of the top-level netlist instances to help visualize the connectivity flow, as shown below.

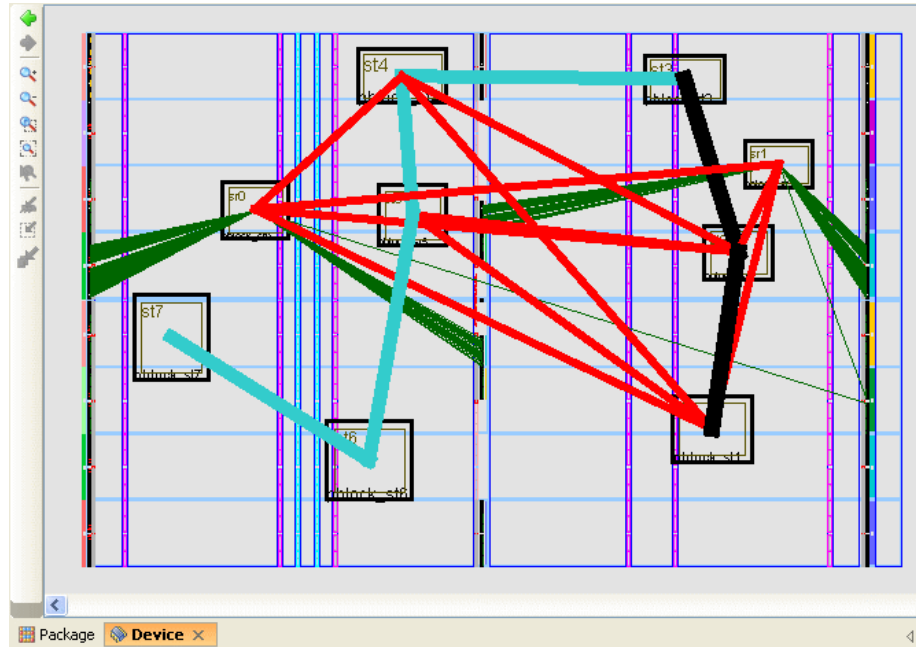


Figure 8-46: Viewing Top-level Design Connectivity

The Net bundles clearly indicate the heaviest connectivity requirements between the modules. The color and line thickness of the Net bundles can be configured depending on the number of signals in them. Selecting a Net bundle will display information about the net content in the Net Bundle Properties view.

You can also traverse down the hierarchy and create submodules for the larger top-level instances for more detailed granularity.

This top-level floorplan can be a good indicator of the quality of the I/O pinout configuration and can help identify potential routing congestion issues. Resource statistics and clock requirements for each module can also be examined to understand potential placement issues.

Refer to the [“Using the Automatic Pblock Commands,”](#) page 311 for more information about creating a top-level floorplan.

Running Design Rule Checks (DRC)

The PlanAhead software contains a set of batch DRC commands that can help verify design integrity prior to running the ISE software. The rules are categorized by type of logic checks being performed.

Running I/O Port and Clock Logic DRCs

Many of the DRC rules are related to I/O pin assignment and clock logic. Refer to [Chapter 5, “I/O Pin Planning”](#) for information on running I/O Port and clock logic related DRCs.

Running Netlist and Floorplan DRCs

1. Select one of the following commands to run the DRC checks on the design:
 - ◆ **Tools > Run DRC**
 - ◆ Select the **Run Design Rule Checker** toolbar button



Figure 8-47: Run Design Rule Checker Toolbar Button

The Run DRC dialog box will appear.

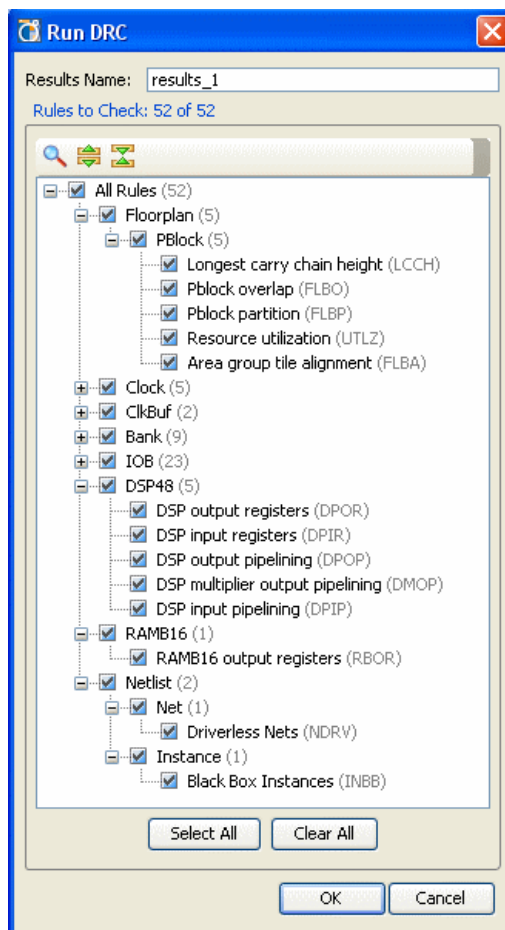


Figure 8-48: Run DRC Dialog Box: Netlist and Floorplan Rules

2. View or edit the Results Name field. Enter a name for the results for a particular run for easier identification during debug in the DRC Violations browser. The output file name will match the name entered.
3. In the Rules to Check group box, use the check boxes to select the design rules to check for each design object. For a description of each rule, see [“DRC Rule Descriptions.”](#)
 - ◆ Expand the hierarchy using the **Expand All** toolbar buttons, or click the **+** next to each category or design object.
 - ◆ Click the check box next to the design object to run all DRCs, click individual DRCs to run individual ones, or click **All Rules** to run a complete DRC.

- Click **OK** to invoke the selected DRC checks.

Viewing DRC Violations

Once completed, the DRC Results view will appear.

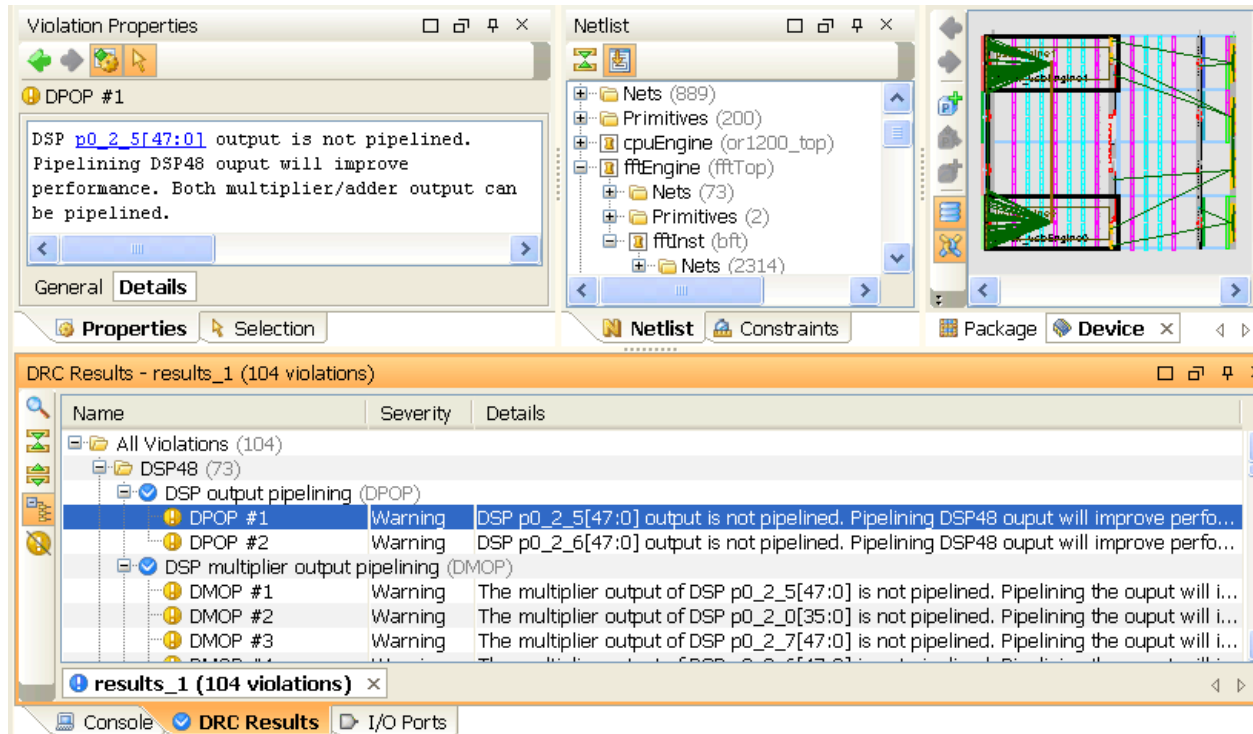


Figure 8-49: DRC Results

Each violation is expanded in the DRC Results view. Errors display a red icon. Warnings display an amber icon. Informational messages display a yellow icon.

By default all errors and warnings are displayed. Click the Hide Warning and Information Messages toolbar button to hide all warnings and info messages and view only errors. Click the toolbar button again to view all errors and warnings once again.



Figure 8-50: Hide Warning and Information Messages Button

Select an error in the DRC Results view list, and the specific violation information is displayed in the Properties view. Select a blue link in the Properties view to highlight the violating design elements in the Device view, Netlist view and Schematic view.

Violations will no longer be displayed in the DRC Results view after the error condition is rectified and the DRC check is rerun.

Each time the Run DRC command is run and errors are detected, a new results tab is added to the DRC Results view. A separate results output file is also created in the PlanAhead invocation directory.

DRC Rule Descriptions

The following tables describe the DRC rules, rule intent and severity.

- [“Floorplan Pblock Rules”](#)
- [“Bank Rules”](#)
- [“DCI Rules”](#)
- [“ClkBuf Rules”](#)
- [“DSP48 Rules”](#)
- [“RAMB16 Rule”](#)
- [“Netlist Rules”](#)

Note: For Global Clock Rules, IOB Rules, and Bank IO Standard Rules, see [“I/O Port and Clock Logic DRC Rule Descriptions.”](#)

Floorplan Pblock Rules

Table 8-3: Floorplan Pblock Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
Longest Carry Chain Height	LCCH	Checks that the Pblock height will accommodate longest Carry Chain assigned to Pblock.	Amber Warning
Pblock overlap	FLBO	Checks for overlapping Pblock rectangles.	Information
Pblock Partition	FLBP	Checks that the LUT to MUXCY and MUXFx connection is not broken by a Pblock partition.	Error
Resource Utilization	UTLZ	Checks that the Pblocks have enough resources for logic assigned to them.	Warning for SLICE logic Error for non-SLICE logic
Area Group Tile Argument	FLBA	Checks that the site ranges in AREA_GROUP constraints are aligned with the CLB grid.	Warning

Bank Rules

Table 8-4: DCI Cascade Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
DCI Cascade Checks	DCIC	Checks that DCI cascade constraint is legal.	Error

Table 8-5: IDelay Control Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
IDelayCtrl Checks	IDLYCTRL	Checks that IDelay placement is consistent with IDlyController locs.	Error

For Bank IO Standard Rules, see [“Bank I/O Standard Rules.”](#)

DCI Rules

Table 8-6: DCI Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
DCI Cascade with part compatibility	DCICPC	Warns the user to load the UCF file into other compatible parts, and to run DRC manually to ensure the DCI cascades are valid.	Warning

ClkBuf Rules

Table 8-7: IDelay Control Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
BufR & BufIO Locations	BUFRIOC	Checks that BUFR and BUFIO driven by the same regional clock terminal are placed at mutually routable locations.	Error

For IOB Rules, see [“IOB Rules.”](#)

DSP48 Rules

Table 8-8: DSP48 Rules

Rule Name	Rule Abbrev	Rule Intent	Severity
DSP output registers	DPOR	DSP48 has a register on the output side; to use this register the register should be synchronously controlled. (Virtex-4)	Information
DSP input registers	DPIR	DSP48 has a register on the input side; to use this register the register should be synchronously controlled. (Virtex-4)	Information
DSP output pipelining	DPOP	DSP48 has a register on the output side; using this pipeline mechanism will improve performance. (Virtex-4)	Information
DSP multiplier output pipelining	DMOP	DSP48 output is not pipelined. Pipelined output will improve performance.	Warning
DSP input pipelining	DPIP	DSP48 has a register on the input side; using this pipeline mechanism will improve performance. (Virtex-4)	Information

RAMB16 Rule

Table 8-9: **RAMB16 Rule**

Rule Name	Rule Abbrev	Rule Intent	Severity
RAMB16 output registers	RBOR	RAMB16 has a register on the output side; to use this register, the register should be synchronously controlled. (Virtex-4)	Information

Netlist Rules

Table 8-10: **Net Rules**

Rule Name	Rule Abbrev	Rule Intent	Severity
Driverless Nets	NDRV	Checks that each net has a proper driver pin.	Warning

Table 8-11: **Instance Rules**

Rule Name	Rule Abbrev	Rule Intent	Severity
Black Box Instances	INBB	Checks that there is no blackbox (undefined logics in the netlist).	Warning

Running Timing Analysis

About TimeAhead

TimeAhead is used in various modes during different stages of design completion. It can provide early estimations of path delays to assist during floorplanning, as well as for detailed path tracing, debugging and constraint assignment. It has two different modes for Estimated and No Interconnect-style analysis. The more physical constraints, such as Pblocks and placement constraints, are assigned in the Floorplan, the more accurate the analysis results will be.

Running TimeAhead

- Run a timing analysis using one of the following methods:
 - ♦ Select **Tools > Run TimeAhead**.
 - ♦ Click the **Run TimeAhead** toolbar button.



Figure 8-51: **Run TimeAhead Toolbar Button**

The Timing dialog box displays.

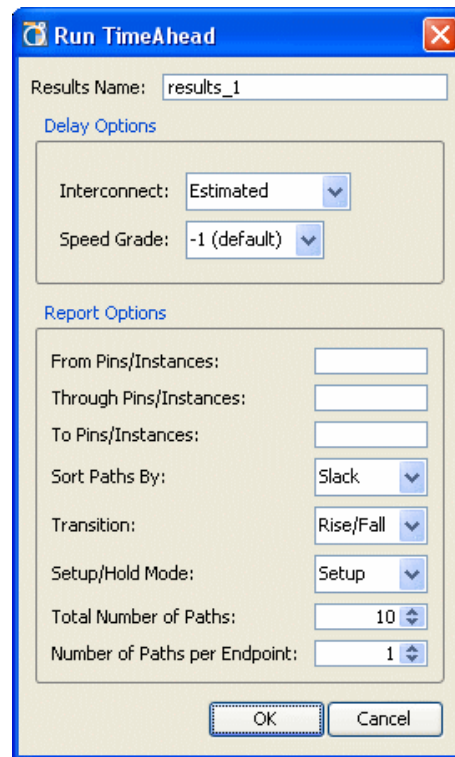


Figure 8-52: Run TimeAhead Dialog Box

2. View or set the definable fields in the Run TimeAhead dialog box:
 - ◆ **Results Name**—View or enter a name for the internal results displayed in the Timing Results view tabs.
 - ◆ **Interconnect**—Set the timer mode used. The field toggles between the following options.
 - **Estimated**—Estimates logic and typical route delays based on pseudo placement.
 - **None**—Estimates all logic primitive delays without any estimated route delays. This report should correlate with the Trace report after the ISE **map** command is run.
 - ◆ **Speed Grade** - Adjust the device speed grade to experiment with alternate speed grades.
 - ◆ **Filter Options** are used to trim the list of paths the Timer will run on. A wildcard (*) can be used to delineate multiple signals.
 - **From Pins/Instances**—is used to select paths that source from this pin or instance name.
 - **Through Pins/Instances**—is used to select paths that travel through this pin or instance name.
 - **To Pins/Instances**—is used to select paths that end at this pin or instance name.
 - **Sort Paths By**—is used to sort the Timing report by constraint group or by slack.

- **Transition**—is used to select either rise or fall.
 - **Setup/Hold Mode**—is used to select Hold analysis versus Setup analysis.
 - **Number of Paths per Clock Group**—is used to determine how many paths to list per constraint.
 - **Number of Paths per Endpoint**—is used to control the number of paths reported per path endpoint.
3. After selections are made, click **OK** to launch TimeAhead.

Excluding Paths from TimeAhead Analysis

TimeAhead can exclude objects from timing analysis by using special control characters in the Filter options fields.

Exclusions

Append an exclamation mark (!) to any name (full name or wildcard name) to signify an exclusion.

For example, if you enter the following:

From Pins/Instances: **pblock_1/***

To Pins/Instances: **!pblock_1/***

The TimeAhead analysis uses the paths starting from instances in pblock_1 but not ending at instances in pblock_1.

Inclusions and Exclusions

You can specify a mix of includes and excludes.

For example, if you enter the following:

From Pins/Instances: **pblock_1/***

To Pins/Instances: **pblock_2/* !pblock_3/***

If pblock_2 and pblock_3 have instances in common (this can happen if pblock_3 is within pblock_2) then paths ending at common instances will not be reported.

Complex Scenarios

Complex scenarios are guided by the following rule of thumb:

If includes are specified then excludes will take away objects from includes. If only excludes are specified then includes are assumed to be all leaf-level instances in the netlist.

Analyzing Timing Results

The Timing Results view can be populated with timing results from either TimeAhead or from the ISE *trce* timing analysis tools. Either TimeAhead or Trce must be run first in order to populate the Timing Results view with paths. The Timing Results view contains the paths that meet the criteria defined in the Run TimeAhead dialog box, as described in “Running Timing Analysis.” When Trce results are imported or when TimeAhead is run with the default options, paths are sorted and listed by constraint.

The Timing Results view is displayed once TimeAhead completes or Trce results are imported.

Name	Type	Slack	From	To	Total Delay	Logic Delay	Net %	Stages
Path 1	Setup	-0.254	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine1/u4/dout[6]	4.054	1.416	65.072	6
Path 2	Setup	-0.252	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine1/u4/dout[22]	4.052	1.418	65.005	6
Path 3	Setup	-0.250	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine1/u4/dout[20]	4.050	1.415	65.062	6
Path 4	Setup	-0.225	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine1/u4/dout[23]	4.025	1.407	65.043	6
Path 5	Setup	-0.179	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem...	3.979	1.630	59.035	4
Path 6	Setup	-0.166	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem...	3.966	1.348	66.011	4
Path 7	Setup	-0.148	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem...	3.961	1.630	58.849	4
Path 8	Setup	-0.155	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine0/u4/dout[6]	3.955	1.416	64.197	6
Path 9	Setup	-0.126	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine0/u4/dout[23]	3.926	1.407	64.162	6
Path 10	Setup	-0.086	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_36/fblk/ins...	usbEngine1/u4/dout[21]	3.886	1.415	63.587	6

Figure 8-53: TimeAhead Timing Results

The interface enables you to examine, sort and select specific paths and instances.

In the Timing Results view, the following information is displayed for each path:

- Constraint Name—Displays the constraint name for the paths listed.
 - ◆ Name—A sequentially number that can be used to sort back to the original order.
 - ◆ Type—Displays whether the path is Setup or Hold related.
 - ◆ Slack—Displays the total positive or negative slack on the path.
 - ◆ From—Displays the path source pin.
 - ◆ To—Displays the paths destination pin.
 - ◆ Total Delay—Lists the total estimated delay on the path.
 - ◆ Logic Delay—Lists the delay attributed to logic delay only.
 - ◆ Net%—Displays the percentage of the delay attributed to routing interconnect.
 - ◆ Stages—Displays the total number of instances on the path including the source and destination which both contribute to the overall delay. This may be different than the method used to calculate levels of logic in ISE.

Note: In TimeAhead, a carry chain interconnect is counted as individual stages of logic.

Sorting the Timing Report

The Timing Results can be sorted by clicking any of the column headers. For example, click on the Slack column header to sort the list by slack. Click the Slack column header a second time to reverse the sort order.

You can sort by a second column by pressing the **Ctrl** key and clicking a second column header. You can sort by as many columns as necessary to refine the list order.

Name	Type	Slack	From	To	Total Delay	Logic Delay	Net %	Stages
Path 3	Setup	-0.250	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine1/u4/...	4.050	1.415	65.062	6
Path 4	Setup	-0.225	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine1/u4/...	4.025	1.407	65.043	6
Path 5	Setup	-0.179	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine0/us...	3.979	1.630	59.035	4
Path 6	Setup	-0.166	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine0/us...	3.966	1.348	66.011	4
Path 7	Setup	-0.148	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine0/us...	3.961	1.630	58.849	4
Path 8	Setup	-0.155	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine0/u4/...	3.955	1.416	64.197	6
Path 9	Setup	-0.126	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine0/u4/...	3.926	1.407	64.162	6
Path 10	Setup	-0.086	usbEngine1/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgffifo18_3...	usbEngine1/u4/...	3.886	1.415	63.587	6

Figure 8-54: Timing Results Sorted by Slack and Total Delay

Click **Ctrl** and click the column header again to remove a sort from a column.

Flattening the List of Paths

By default, the paths are categorized by constraint. You can flatten the list and view all paths by selecting the **Group by Constraint** in the Timing Results view icon in the Timing Results view toolbar.



Figure 8-55: Group Paths by Constraint Toolbar Button

The Group by Constraint toolbar button toggles between a categorized list of constraints and a flattened list of paths.

Removing Paths from the Timing Report

Paths can be selectively removed from the timing report to make for easier sorting and viewing of critical paths.

1. Select the paths to remove paths from the timing report. To select multiple paths, use the **Shift** or **Ctrl** keys, and select the paths.
2. Press the **Delete** key, or select **Delete** from the popup menu in the Timing Results view.

Displaying Path Details

When a path is selected from the list, the Path Properties view is populated with information about the path. All logic elements are listed with detailed delay information and hyperlinks.

The screenshot shows the 'Path Properties' window with the 'Report' tab selected. It displays detailed delay information for 'Path 7'. The window is divided into three main sections: Summary, Source Clock Path, and Destination Clock Path. Each section contains a table with columns for Delay Type, Delay, Cumulative, Location, PBlock, and Logical Resource.

Summary

Name	Path 7				
Slack	-0.161				
Source	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgifo18_36/fblk/inst_few/k1[1].inst_fed/one_prim.inst_fifoprim/gfiffo36.sngfiffo36/fifo36_wrap_inst/DOP[1]				
Destination	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem_generator/valid.cstr/ramloop[0].ram.r/v5.ram/SP.WIDE_PRIM18.SP				
Requirement	3.800				
Delay	3.961				
Source Clock	usbClk (rising at 0.000ns)				
Destination Clock	usbClk (rising at 3.800ns)				

Source Clock Path

Delay Type	Delay	Cumulative	Location	PBlock	Logical Resource
	0.000	0.000	AB19		usbClk
net (fo=0)	0.000	0.000	AB19		usbClk_ibuf/ibufg/I
IBUFG	0.818	0.818	AB19		usbClk_ibuf/ibufg/O
net (fo=1)	0.000	0.818	BUFGCTRL_X0Y9		usbClk_ibuf/ibufg/I
BUFG	0.250	1.068	BUFGCTRL_X0Y9		usbClk_ibuf/ibufg/O
net (fo=407)	2.033	3.101	RAMB36_X2Y1	pblock_usbEngine0	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgifo18_36/fblk/inst_few/k1[1].inst_fed/one_prim.inst_fifoprim/gfiffo36.sngfiffo36/fifo36_wrap_inst/DOP[1]
Total	3.101	3.101			

Data Path

Delay Type	Delay	Cumulative	Location	PBlock	Logical Resource
FIFO36_EXP	0.818	0.818	RAMB36_X2Y1	pblock_usbEngine0	usbEngine0/usb_dma_wb_in/BU2/U0/gen_fifo18_36.fgifo18_36/fblk/inst_few/k1[1].inst_fed/one_prim.inst_fifoprim/gfiffo36.sngfiffo36/fifo36_wrap_inst/DOP[1]
net (fo=40)	1.520	2.338	SLICE_X19Y8	pblock_usbEngine0	usbEngine0/u2/wsel/I0
LUT6	0.094	2.432	SLICE_X19Y8	pblock_usbEngine0	usbEngine0/u2/wsel/O
net (fo=42)	0.381	2.813	SLICE_X18Y9	pblock_usbEngine0	usbEngine0/u2/sram_we/I14
LUT6	0.094	2.907	SLICE_X18Y9	pblock_usbEngine0	usbEngine0/u2/sram_we/O
net (fo=4)	0.430	3.337	RAMB36_X1Y1	pblock_usbEngine0	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem_generator/valid.cstr/ramloop[0].ram.r/v5.ram/SP.WIDE_PRIM18.SP
RAMB18	0.624	3.961	RAMB36_X1Y1	pblock_usbEngine0	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem_generator/valid.cstr/ramloop[0].ram.r/v5.ram/SP.WIDE_PRIM18.SP
Total	3.961	3.961			

Destination Clock Path

Delay Type	Delay	Cumulative	Location	PBlock	Logical Resource
	0.000	0.000	AB19		usbClk
net (fo=0)	0.000	0.000	AB19		usbClk_ibuf/ibufg/I
IBUFG	0.818	0.818	AB19		usbClk_ibuf/ibufg/O
net (fo=1)	0.000	0.818	BUFGCTRL_X0Y9		usbClk_ibuf/ibufg/I
BUFG	0.250	1.068	BUFGCTRL_X0Y9		usbClk_ibuf/ibufg/O
net (fo=407)	2.033	3.101	RAMB36_X1Y1	pblock_usbEngine0	usbEngine0/usbEngineSRAM/BU2/U0/blk_mem_generator/valid.cstr/ramloop[0].ram.r/v5.ram/SP.WIDE_PRIM18.SP
Total	3.101	3.101			

The window also includes tabs for 'General', 'Report' (selected), 'Instances', and 'Options'. At the bottom, there are buttons for 'Properties' and 'Selection'.

Figure 8-56: Path Properties: Report tab

Notice how the report has a similar format to the Trce report.

By default, selecting a path will also select all instances contained in it. Selecting any of the blue-hyperlinked objects in the report will also select them in other views such as the Netlist and Device views.

Multiple paths may be selected using the **Shift** and **Ctrl** keys. All instances contained in all selected paths will be selected, but the Path Properties will only display information about the first path selected.

Displaying Timing Path Reports in the Workspace

An individual Timing Path Report can also be displayed in the Workspace for easier viewing. To view the report:

1. Select the desired timing path.
2. Click the **View Path Report** popup menu command, or click the **View path timing report in the workspace** toolbar button in the Timing Results view toolbar.



Figure 8-57: **View Path Timing Report in the Workspace Button**

Searching for Objects Using the Find Command

PlanAhead enables you to selectively search for instances or nets using the Find command. To invoke the Find command:

1. Select **Edit > Find**, or click the **Find** toolbar button.



Figure 8-58: **Find Toolbar Button**

The Find dialog box will appear.

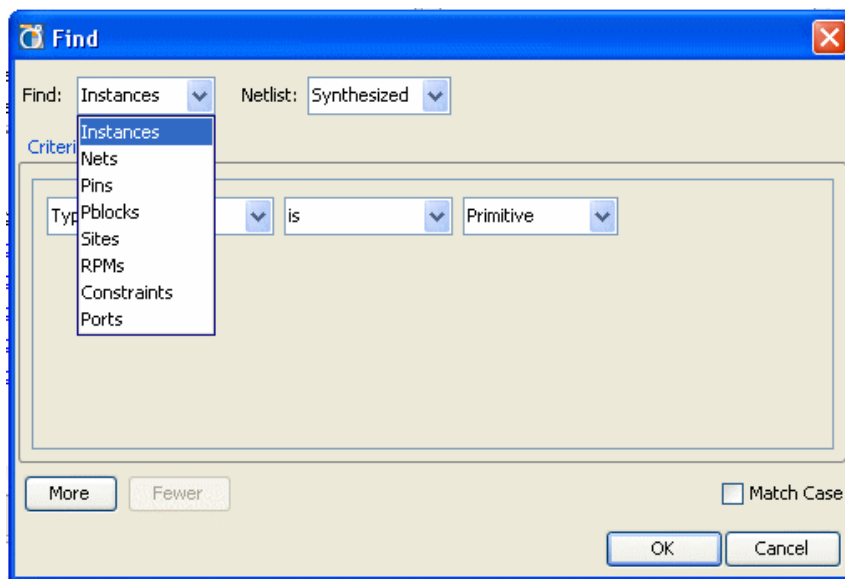


Figure 8-59: **Find Dialog Box**

2. View or edit the definable fields in the Find dialog box:
 - ♦ **Find**—Select the object type (Instances, Nets, Pblocks, etc.) you wish to search for.
 - ♦ **Netlist**—Select whether to search the RTL or Synthesized netlist
 - ♦ **Criteria**—For each object type, a different set of search parameters are available in the dialog box.

In the first field, select the way in which you would like to search for the objects: Name, Status, Type, Parent Pblock, Module, or Primitive count.

The second field can be used to set boolean options for the search, such as matches, doesn't match, contains, or doesn't contain.

The third field enables you to input search criteria strings. An asterisk (*) can be used to define the search strings.

- Optionally, click the **More** button to define additional search filters or to simultaneously search for additional types of objects. A new row of search criteria fields will display in the Find dialog. An AND/OR field appears to define the additional search criteria. Setting it to AND will define an additional search filter as displayed below.

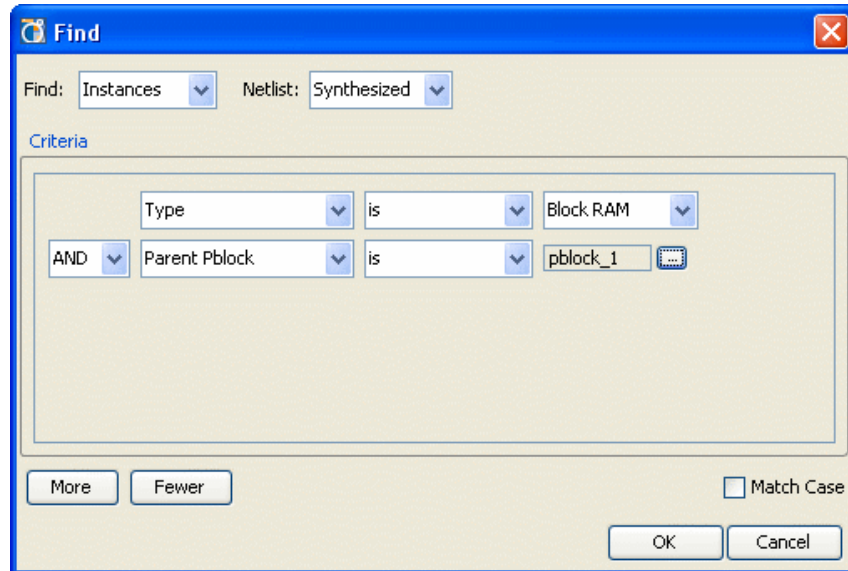


Figure 8-60: Searching for Objects with Additional Search Criteria

- Multiple objects can be searched for simultaneously by using the OR criteria as shown below.

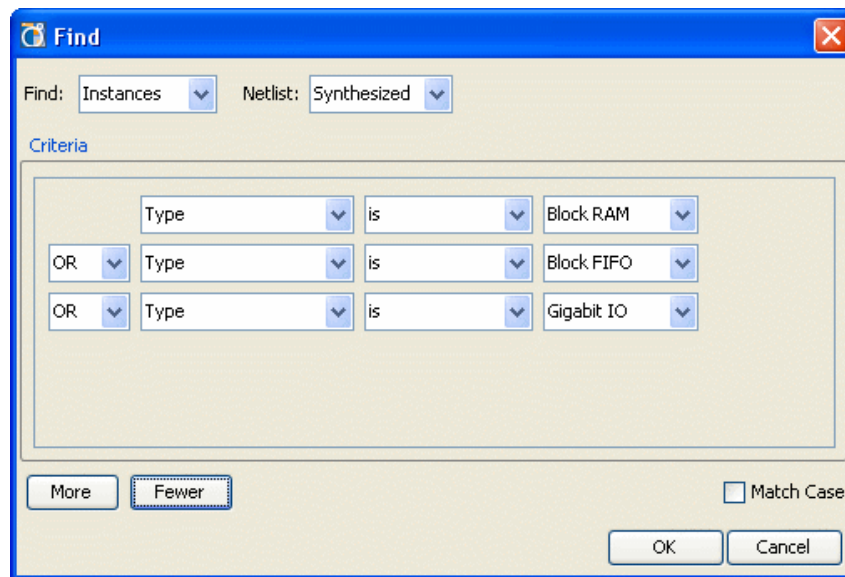


Figure 8-61: Searching for Multiple Object Types Simultaneously

5. Click the **Fewer** button to remove search criteria rows.
6. Click **OK** to perform the search.

The combined search results are displayed in the Find Results view.

Using the Find Results View

The objects matching the Find dialog box criteria are displayed in the Find Results view once you initiate the search by clicking **OK**.

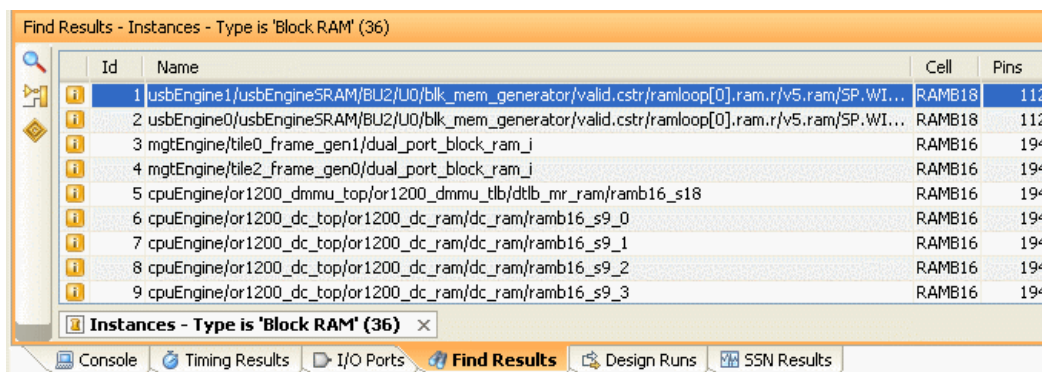


Figure 8-62: Find Results View

A new Find Results tab is created each time the Find command is run. The tab is named according to the Search criteria and number of objects found.

Objects can be selected directly from the Find Results dialog box. Selecting objects from the list of found objects will select them in other PlanAhead views. Multiple elements can be selected by using the **Shift** or **Ctrl** keys. Additional commands are available using the popup menu.

The Find Results can be sorted by clicking any of the column headers. You can sort by a second column by pressing the **Ctrl** key and clicking a second column header.

Click the **X** in a Find Results tab to close Find Results views.

Analyzing Implementation Results

This chapter contains the following sections:

- [“Importing ISE Placement and Timing Results”](#)
- [“Analyzing Placement and Timing Results”](#)
- [“Exploring Logic Connectivity”](#)
- [“Highlighting Selected Objects”](#)
- [“Highlighting Placed Modules”](#)
- [“Marking Selected Objects”](#)
- [“Displaying Design Metrics”](#)

Importing ISE Placement and Timing Results

Importing ISE Placement and Timing Results With New Project Wizard

You can create a new project and populate the project with a netlist, and timing and placement report files from an ISE® software implementation. For more information about creating a new project that imports ISE implementation results, see [“Creating a Project with ISE Placement and Timing Results.”](#)

Importing ISE Placement and Timing Results from PlanAhead Runs

ISE placement and timing results can be imported and used to analyze the design. The PlanAhead™ environment makes it easy to import the results from any PlanAhead run.

For more information about implementing the design, see [“Importing Run Results.”](#)

Importing ISE Placement and Timing Results from Outside PlanAhead

Implementation results from outside of PlanAhead can be imported using the **File > Import Placement** and **File > Import TRCE Results** commands. For more information about importing ISE implementation results, see [“Importing ISE Implementation Results.”](#)

Importing ISE Placement and Timing Results from Project Navigator

When using PlanAhead with Project Navigator, ISE placement and timing results are automatically imported when you run the **Analyze Timing/Floorplan Design** process in Project Navigator. Additional ISE placement or trace timing results from previous runs can be imported by using the PlanAhead **File > Import Placement** or **File > Import TRCE Results** command.

For more information, see [Chapter 3, “Using PlanAhead With Project Navigator.”](#)

Analyzing Placement and Timing Results

Exploring Xilinx TRCE Results

PlanAhead enables you to import timing reports (TWX or TWR) generated by the Xilinx® **trce** command. Once imported, all signal tracing and viewing capabilities, which include path selection, highlighting and tracing, are available for analyzing the trce results. In the Timing Results view, each imported timing report is displayed in a separate tab; each tab displaying the user-specified results display name.

PlanAhead provides links to data sheets for more information about timing delay name parameters. PlanAhead:

- Links the timing report delay name parameters to the appropriate device data sheets.
- Enables you to search data sheets for the matching timing delay name parameter, and displays the data sheet search results.

Note: If the searched timing delay name parameter is not available in any data sheet, PlanAhead indicates that no results are found.

For more information about importing trce results, refer to [“Importing Run Results”](#) and [“Importing ISE TRCE Timing Results into an Existing Project.”](#)

Viewing Timing Paths in the Device View

You can view timing paths in the Device view (in the Workspace) when you select a path row or rows in the Timing Results view. The path is highlighted in the Device view. Multiple paths can be selected and all instances found in the path will be selected and highlighted.

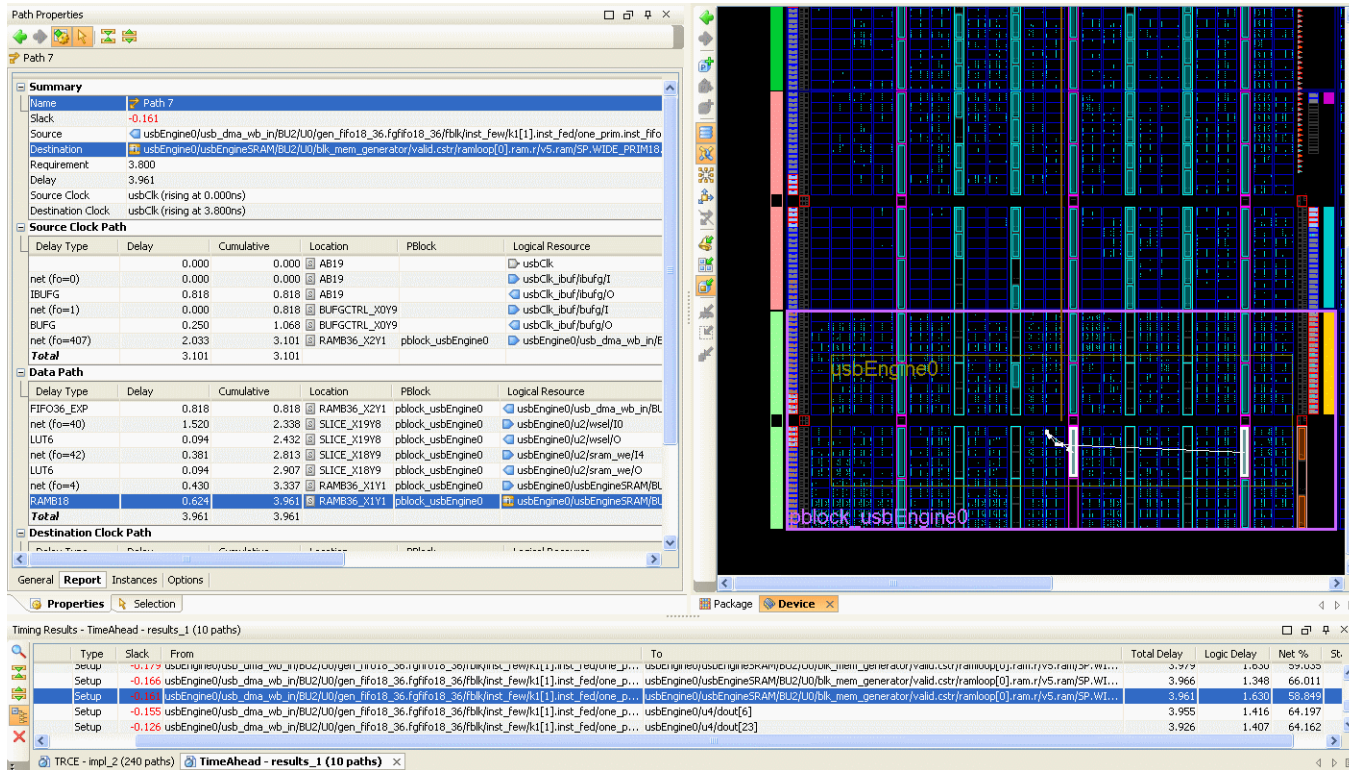


Figure 9-1: Highlighted Timing Paths in the Device View

Viewing Timing Paths in Schematic View

If you select **Schematic** from the Timing Results popup menu, PlanAhead will generate a Schematic view displaying all of the instances found in the selected paths. The Schematic view clearly displays the instances along with the hierarchical modules.

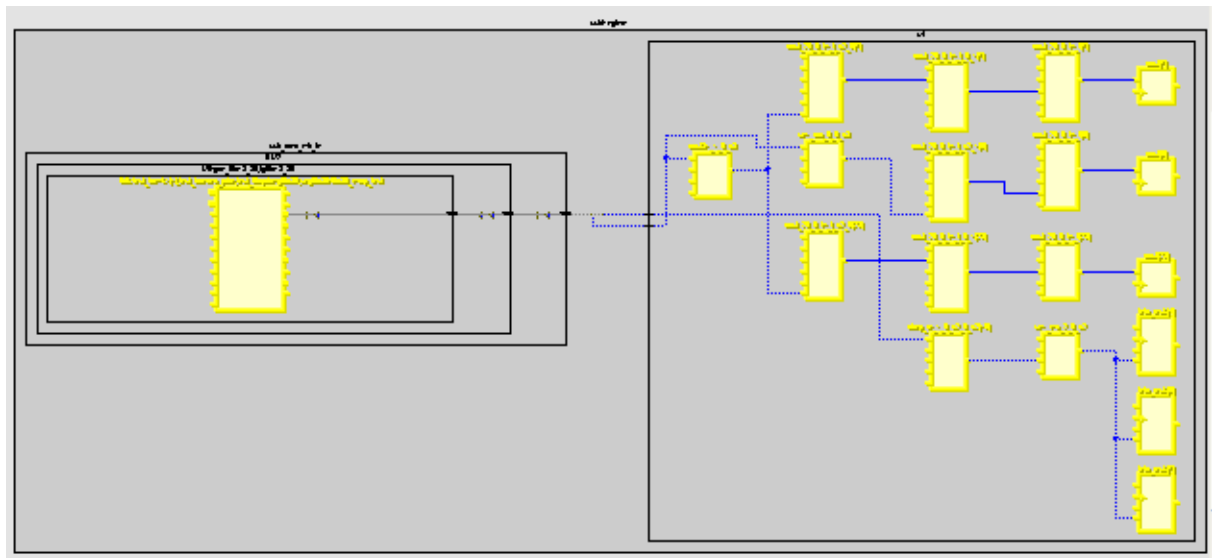


Figure 9-2: Timing Paths Displayed in Schematic View

When the Schematic view is generated on a path, all of the objects are displayed. When a Schematic view for individual logic instances is generated, only the selected instances are displayed.

All of the instances from a group of paths can be displayed in this manner making it easy to identify what modules to be grouped together for floorplanning. The Schematic view Pblock creation popup commands make assignment to Pblocks very straightforward. For more information about Schematic expansion and traversal commands, see [“Using the Schematic View,”](#) page 227 of [“Analyzing the Design.”](#)

Exploring Logic Connectivity

Using the Show Connectivity Command

The Show Connectivity command highlights all of the nets connected to the selected elements. To use this command:

1. Select a net, pblock, instance or combination thereof.
2. Select **Show Connectivity** from the popup menu.

For example, if an instance or Pblock is selected, all of the nets connecting to that element will be highlighted.

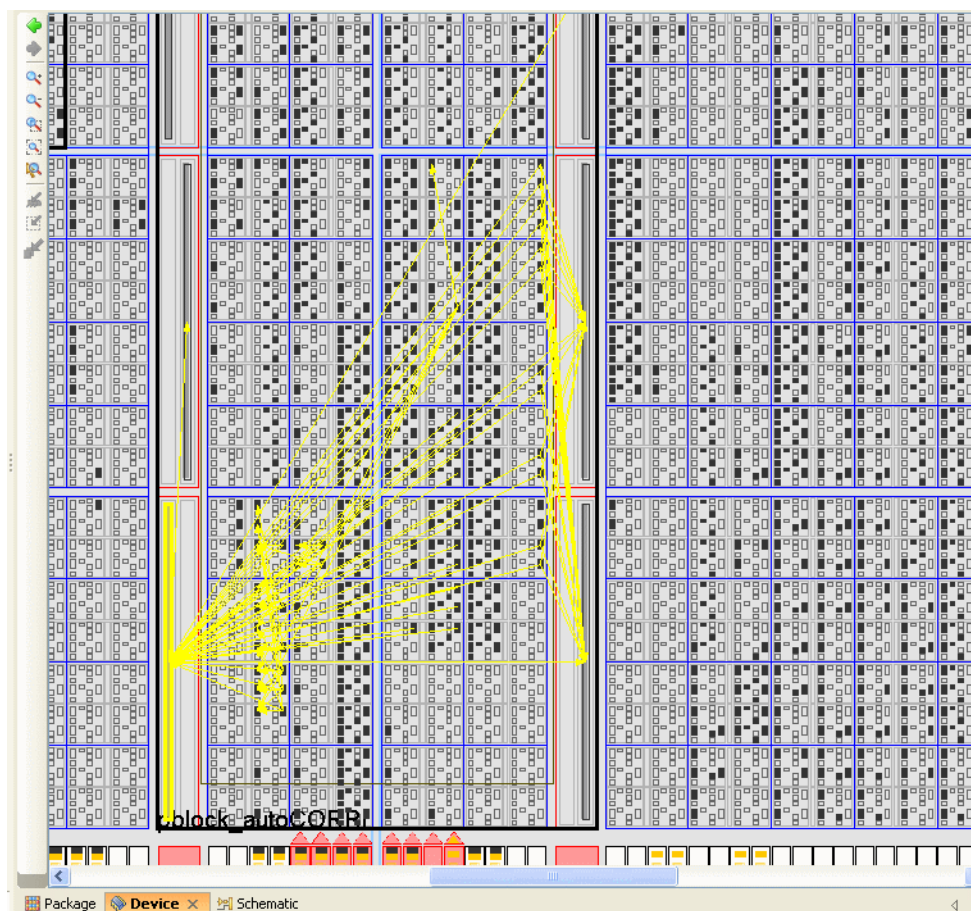


Figure 9-3: Net Connectivity in the Device View

The Show Connectivity command can be continuously run on newly selected objects by toggling the **Show connections for selected instances** toolbar button on and off.



Figure 9-4: **Show connections for selected instances Mode Toolbar Button**

Selecting a Cone of Logic

The Show Connectivity command can be run sequentially to continue to select and expand a logic cone.

1. Select a net, pblock, instance or combination thereof.
2. Select **Show Connectivity** from the popup menu. The command will highlight all nets connected to the selected element, as described above.
3. Select **Show Connectivity** from the popup menu a *second* time. The command will select the set of connected instances to those nets.
4. Select **Show Connectivity** from the popup menu a *third* time. The command will highlight the next level of nets connected to the selected instances, and so on.

This is an easy way to select a cone of logic starting at a particular instance or I/O Port.

Expanding Logic in the Schematic View

Logic can be traced throughout the design hierarchy using the Schematic view. Signals can be interactively expanded by double-clicking on the pins of the instance to be traced.

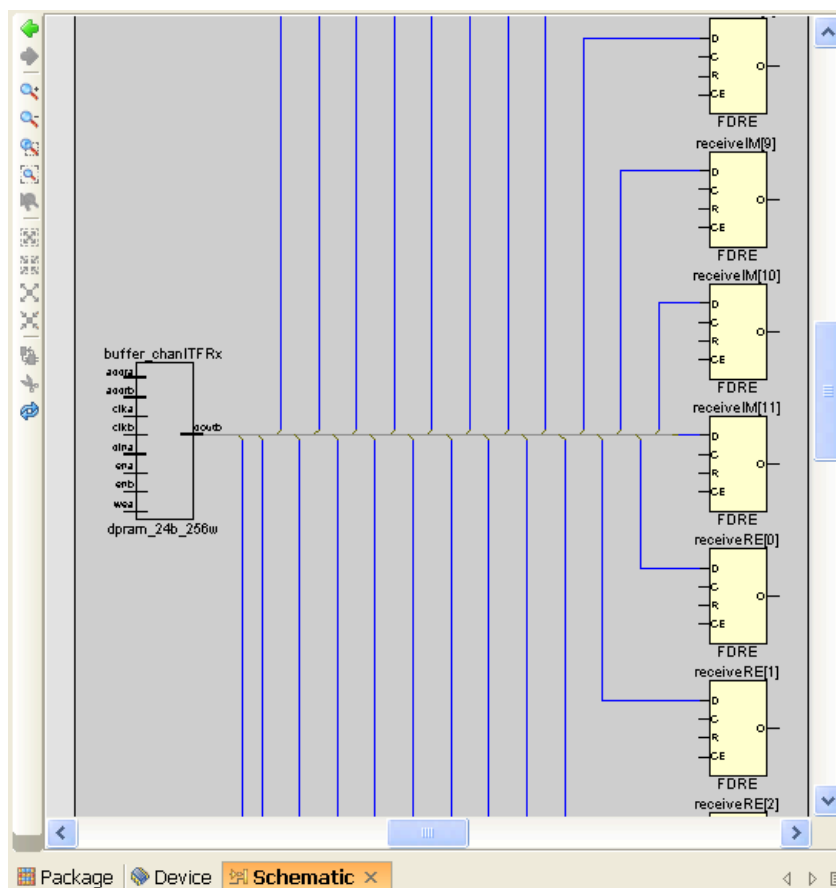


Figure 9-5: Logic Expanded in the Schematic View

Instance and module connectivity, and content can also be interactively expanded and displayed.

Anything selected in the Schematic view will also be highlighted in the Device view. Traced logic is easier to see in the Device view after the implementation placement results have been imported.

For more information about exploring logic in the schematic, see [“Using the Schematic View.”](#)

Tracing Logic Paths using the Properties Dialog Box Connectivity Tabs

The Net Properties and Instance Properties commands launch a Net/Instance Properties window that contains a Connectivity tab. In this tab, selective expansion of logic in the design are displayed.

To begin tracing logic:

1. Select a net or an instance.
The Net Properties or Instance Properties view should appear in the Properties view.
2. If the Net/Instance Properties are not displayed, right-click on the net or instance, and select **Net Properties** or **Instance Properties** from the popup menu.

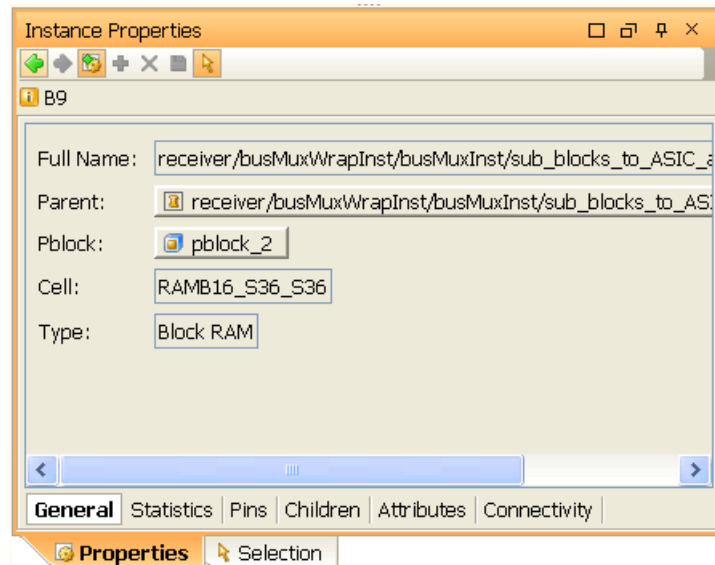


Figure 9-6: Instance Properties

Notice in the example above that the RAMB16 instance is selected. In the Connectivity tab of the Instance Properties view, all of the nets that connect to that instance are listed along with the pin names they connect to. The number of pins on the net is also listed. To view individual nets, simply select a line in the list.

To further trace a particular net:

1. Select the line in the list in the Connectivity tab.
2. Select the **Net Properties** command from the popup menu.

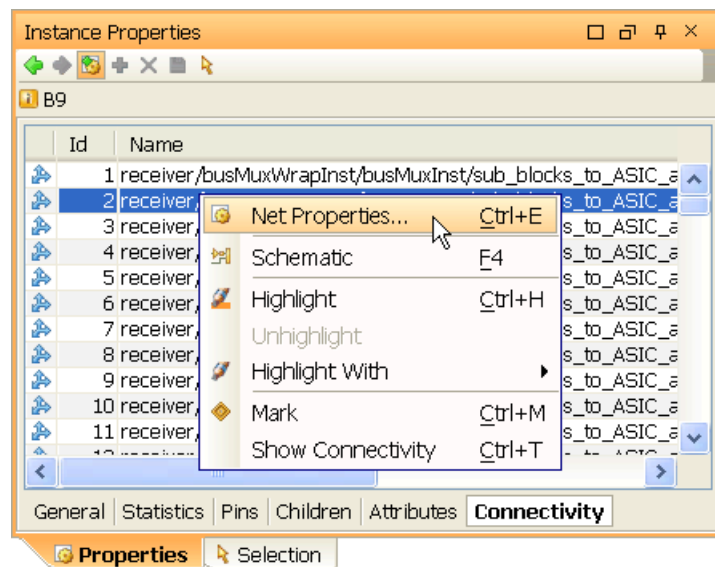


Figure 9-7: Tracing Nets from Connectivity Tab

The Net Properties will then display under the Connectivity tab. All of the instances and pin names the net connects to are listed. You can continue this process to traverse any logic in the design.

To step back or advance through the selected items, click the **Previous Object** and **Next Object** arrow toolbar buttons. The highlighted items displayed will also be recalled as the previous steps are navigated.

Highlighting Selected Objects

Highlighting Objects

PlanAhead has a very flexible highlight mechanism allowing selective highlighting of objects. Highlighting enables you to display multiple placement groups at once using one or more colors. Highlighted objects remain highlighted even when you click elsewhere in PlanAhead. You can highlight any number of selected objects.

PlanAhead has a lot of different ways to select the desired logic objects. Once selected, objects can be highlighted with the **Select > Highlight** command or by selecting **Highlight** from the popup menu in most views. This command operates on the selected logic.

When highlighting Pblock logic, you may use the **Highlight Primitives** command in order to highlight the lower level logic. For more information, see [“Using the Select Primitives and Highlight Primitives Commands.”](#)

Unhighlighting Objects

To unhighlight objects use one of the following commands:

- Choose **Select > Unhighlight All** to unhighlight all objects.
- Choose **Select > Unhighlight Color** to unhighlight based on color.
- Click the **Unhighlight All** toolbar button.



Figure 9-8: Unhighlight All Toolbar Button

Highlighting Placed Modules

Using the Select Primitives and Highlight Primitives Commands

After XDL placement has been imported, use the **Highlight Primitives** command to selectively highlight the underlying primitive logic elements for Pblocks and logic modules. You can select logic modules or Pblocks, and use the Highlight Primitives command and then select a color to highlight their associated placement.

When multiple instances are selected, you can select the same color for all or use **Cycle Colors** to use different highlight colors for each of the selected modules.

Modules and primitives in the Netlist view are marked with the matching highlight color in the Device, Schematic and Package views.

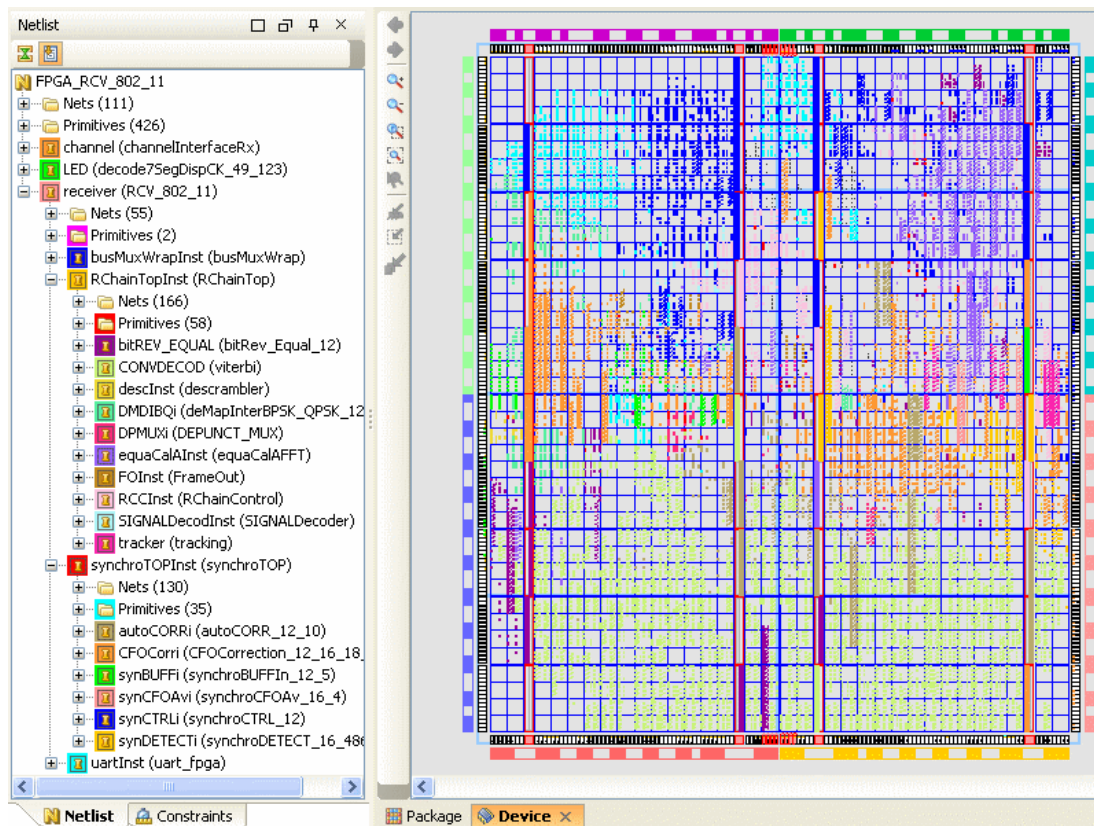


Figure 9-9: Matching Highlighting Color in the Netlist and Device Views

Marking Selected Objects

Marking Objects

PlanAhead enables you to place a Mark symbol in the Device view for all selected objects. Marking a selected object is particularly helpful when displaying small objects that you wish to see in the Device view. To mark selected objects, select **Select > Mark**, or press **Ctrl+M**. This command is available in other views, including the Netlist and Physical Hierarchy views.

When marking timing paths, the start point is marked in green, the end point in red, and all intermediate points in yellow, as shown below.

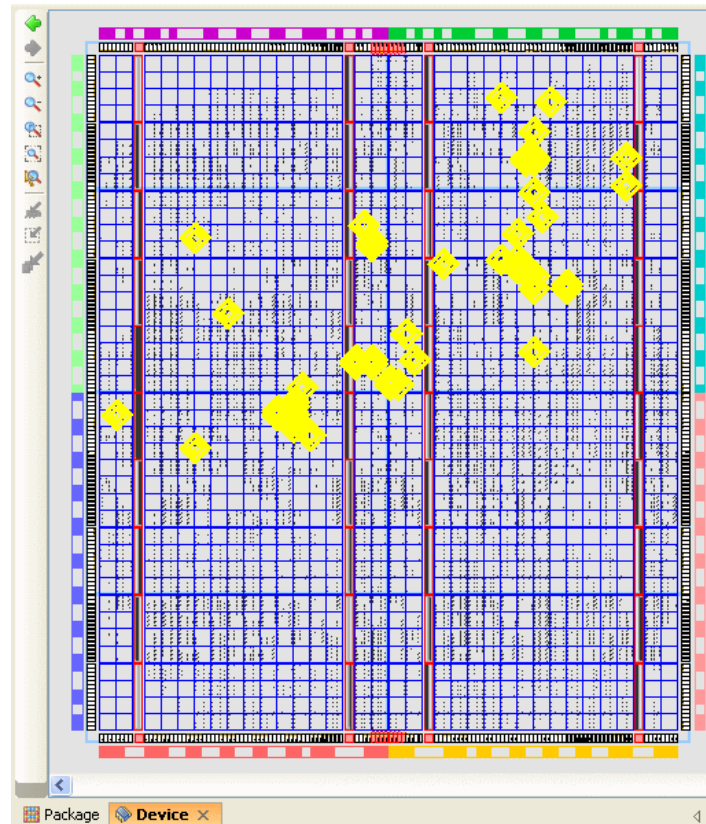


Figure 9-10: Marked Symbols in Device View

Removing Marks

Marks can be removed using one of the following methods:

- Choose **Select > Unmark All**.
- Click the **Unmark All** toolbar button.



Figure 9-11: Unmark All Toolbar Button

Displaying Design Metrics

Using the Metrics View

The PlanAhead Metrics view displays a list of design metrics that can be displayed using a colored graph of the potential problem areas in the design. The current metrics include utilization and timing checks at both the Pblock and placed design level.

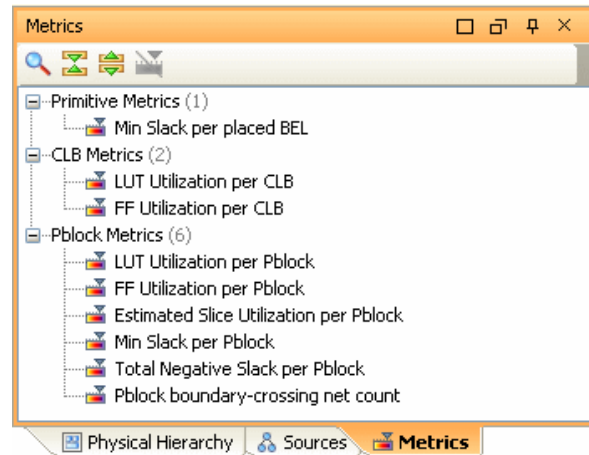


Figure 9-12: Metrics View

The Metric Properties view provides a description of the Metric function along with the bins defined to highlight potential problems, as shown in the example below.

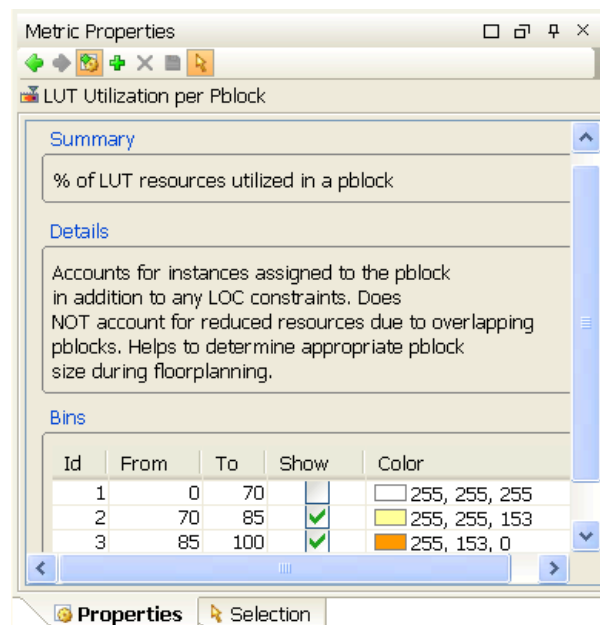


Figure 9-13: Properties View: Set Metric Range Values

Displaying the Metric Maps in the Device View

To display a Metric map in the Device view, select the metric, and select the **Show** popup menu command. A color based metric map is displayed.

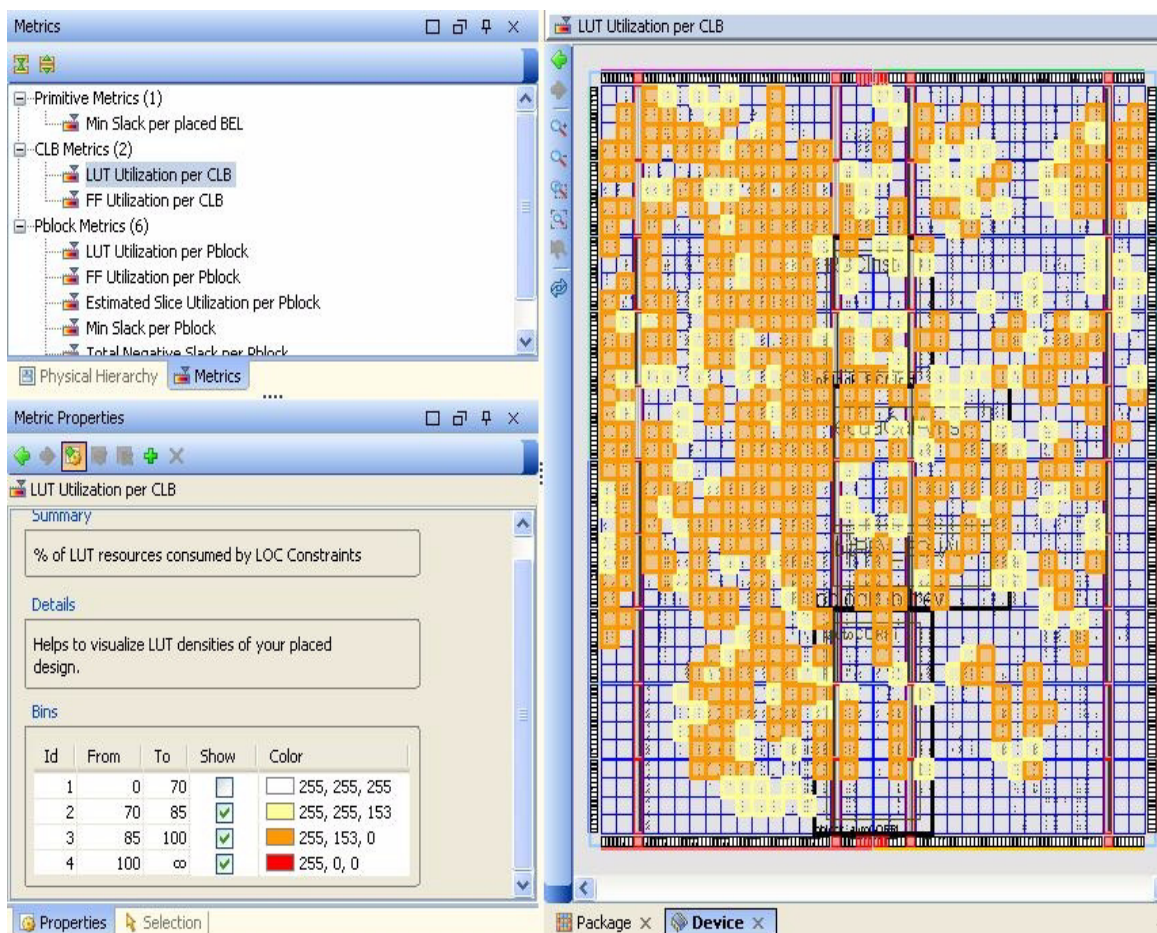


Figure 9-14: Metric Map in the Device View

In order to display any of the slack related metrics, a TimeAhead timing estimated mode analysis must first be run. In order to display any of the CLB or BEL constraints type metrics, the placement results from ISE implementation must be imported. For more information, see [“Importing Placement Results into Existing Project.”](#)

Multiple metric maps can be displayed simultaneously.

Hiding Metric Map Display

To hide a metric map in the Device view, select the metric, and select the **Hide** or **Hide Metrics** from the popup menu, or click the **Hide Metrics** toolbar button.



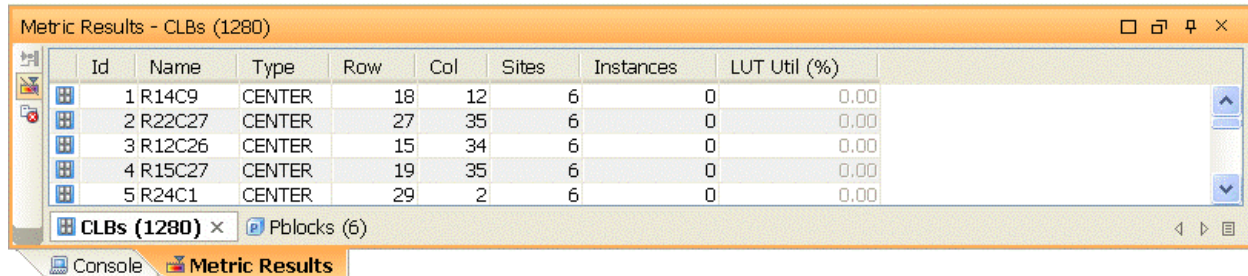
Figure 9-15: Hide Metrics Toolbar Button

Using the Metrics Results View

After you select the **Show** command, the metric results are displayed in the bottom view.

The information in the Metric Results view can be sorted by clicking any of the column headers. You can sort by a second column by pressing the **Ctrl** key and clicking a second column header. Add as many sort criteria as necessary to refine the list order.

The Results are automatically updated as the floorplan is modified. The different types of metrics, such as for Pblocks, CLBs, and primitives, are displayed in different charts. Each type has its own tab along the bottom of the Metrics Results view.



	Id	Name	Type	Row	Col	Sites	Instances	LUT Util (%)
	1	R14C9	CENTER	18	12	6	0	0.00
	2	R22C27	CENTER	27	35	6	0	0.00
	3	R12C26	CENTER	15	34	6	0	0.00
	4	R15C27	CENTER	19	35	6	0	0.00
	5	R24C1	CENTER	29	2	6	0	0.00

CLBs (1280) x Pblocks (6)

Console Metric Results

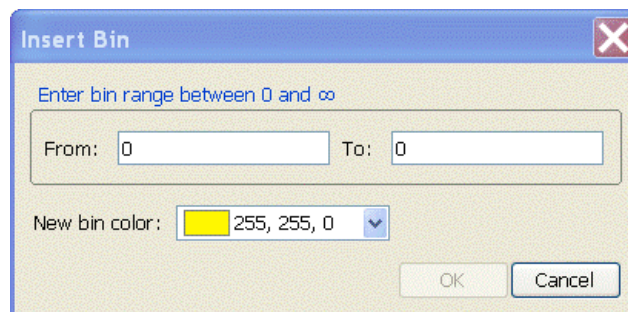
Figure 9-16: Metric Type Tabs in Metrics Results View

Configuring the Metric Ranges

The bin ranges for each map can be configured within the Properties view.

Colors and the range display are fully adjustable. New bins can be added or deleted to define desired ranges. To do so, select the **Apply changes** button in the Metrics Properties view or select **Apply Changes** from the popup menu.

To insert a new range bin, select the desired bin to split, and select **Insert Bin** from the right-click popup menu. The dialog box below allows the definition of the range and color.



Insert Bin

Enter bin range between 0 and ∞

From: 0 To: 0

New bin color: 255, 255, 0

OK Cancel

Figure 9-17: Insert Bin Dialog Box

The ranges are adjusted to accommodate the newly defined range.

Floorplanning the Design

This chapter contains the following sections:

- [“Floorplanning Overview”](#)
- [“Partitioning the Design by Creating Pblocks”](#)
- [“Working with Pblocks”](#)
- [“Using Resource Utilization Statistics to Size Pblocks”](#)
- [“Placing Pblocks Based on Connectivity”](#)
- [“Using the Automatic Pblock Commands”](#)
- [“Working with Placement LOC Constraints”](#)
- [“Using IP Reuse Capabilities”](#)

Floorplanning Overview

The PlanAhead™ software supports a floorplanning methodology that allows designers to constrain critical logic in order to ensure shorter interconnect lengths with less delay. This methodology involves user interaction with the physical design and is not a push-button flow. Designers can use the analysis capabilities in PlanAhead coupled with their own knowledge of the design to define a Floorplan aimed at improved performance.

Floorplanning can be accomplished by creating physical block (Pblock) locations to constrain logic placement or by locking individual logic objects to specific device sites.

Floorplanning Methodology Tips

Below are some helpful floorplanning methodology tips:

1. Use the methods described in the [Chapter 8, “Analyzing the Design”](#) to gain an understanding of the design data flow and the resource requirements of the various modules. Modules with block memory and block arithmetic sites must be taken into consideration since their placement sites can be somewhat restrictive.
2. Where possible, use your knowledge of the design to create Pblocks for critical modules. The analysis from TimeAhead, the Schematic view, and/or place and route results can provide guidance.
3. When floorplanning for performance, it is good practice to only constrain the hierarchies that contain the critical path. In some cases, the hierarchy that is connected to fixed silicon resources (I/Os or PPCs) should be floorplanned as well. Floorplanning all the logic in a design, as one might do in an ASIC flow, will generally hurt performance. Since FPGA tools work differently from ASIC tools, just as FPGA architecture is different from ASIC architecture. It is rare that floorplanning an entire FPGA design will help performance.

4. Use the Pblock Properties' statistics to make the Pblock as small as practical to limit interconnect length. Use these same statistics to ensure that the RPMs and Carry Chains will fit in the Pblock rectangle.
5. Larger Pblocks may need to be partitioned further to provide a finer level of placement constraint granularity. As a rule of thumb keep the size of a single Pblock to less than 30% of the chip. Smaller Pblocks are generally better.

Partitioning the Design by Creating Pblocks

The process of floorplanning begins by partitioning some or all of the logic in the design to group and constrain it, which prevents migration during implementation. PlanAhead provides the ability to hierarchically partition the design into smaller, more manageable physical blocks (Pblocks). PlanAhead maintains a physical hierarchy that is independent from the logic hierarchy. This enables Pblocks to include logic modules and primitive logic from anywhere in the logic hierarchy. Critical or associated logic can be tightly grouped together into a single Pblock, which prevents logic migration, limits interconnect lengths, and reduces delays.

Floorplanning is initiated by creating rectangular Pblocks in the Device view. Pblocks may also be created without rectangles. The ISE® tools will attempt to group the logic together during implementation.

Partitioning can be performed manually by creating Pblocks or automatically by using the Partitioner.

Creating a Pblock will result in an AREA_GROUP constraint being written in the exported UCF constraint file. The logic assigned, ranges specified and attributes defined in PlanAhead will be reflected in the constraints.

Several commands are available to create Pblocks, as described in the following sections.

Using the Draw Pblock Command

The Draw Pblock commands will assign pre-selected logic to a new Pblock in the Device view. You can select the desired logic to assign to the Pblock prior to invoking the command.

To create a Pblock:

1. Select the logic in any view, such as the Netlist view, to assign to the Pblock.
2. Select **Draw Pblock** from the popup menu, or click the **Draw Pblock** toolbar button.



Figure 10-1: Draw Pblock Toolbar Button

3. Move the cursor to the location within the Device view where a Pblock corner is desired.
4. Press and hold the left mouse button, move the mouse cursor to the opposite corner of the Pblock, and release the mouse button.

The New Pblock dialog box will appear.

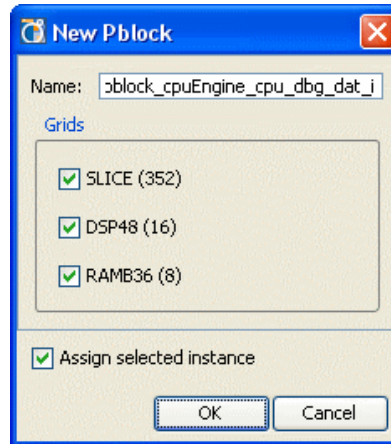


Figure 10-2: New Pblock Dialog Box

5. View and edit the options in the New Pblock dialog box.
 - ◆ **Name**—Enter a suitable name for the Pblock. If no name is entered, a default name of “pblock_n” or “Pblock_<instancename>” will be used.
 - ◆ **Grids**—Select the desired device resource ranges to be constrained by the Pblock
 - ◆ **Assign selected instances**—Select to assign the selected instances to the new Pblock. Occasionally, users inadvertently have logic selected, which is not intended for assignment.
6. Click **OK** to create a new Pblock.

The Pblock will be displayed and selected in the Device view and the Physical Hierarchy view.

The initial Pblock size and location are not critical during manual creation. Pblocks can be appropriately sized and located by using dynamic Resource Utilization Statistics in the Pblock Properties dialog box. Pblocks can be placed appropriately by viewing the connectivity display in the Device view.

Sometimes, it is helpful to initially create all of the Pblocks with small rectangles to help visualize the connectivity flow between the Pblocks prior to attempting sizing.

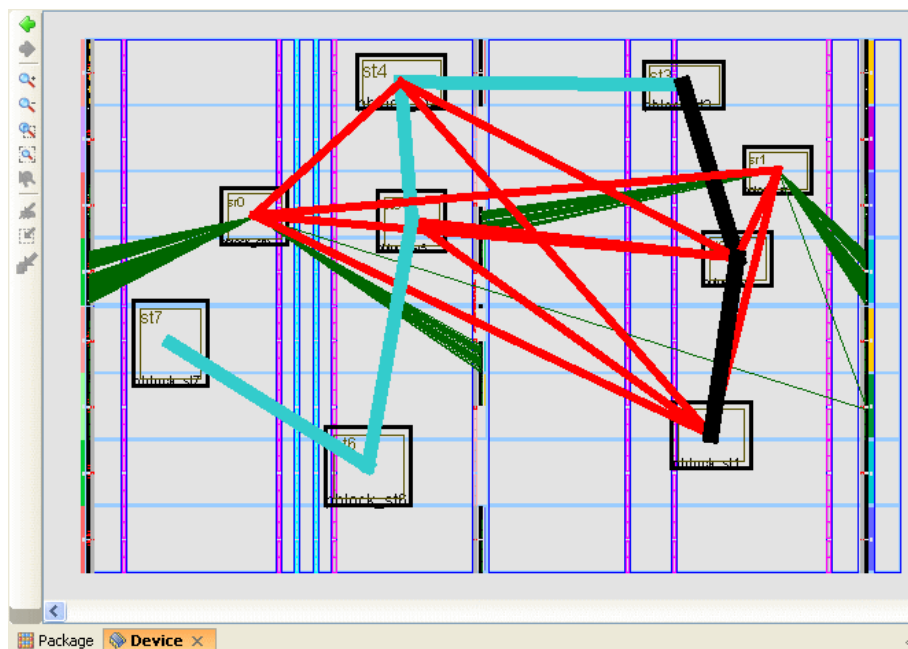


Figure 10-3: Pblocks in the Device View

Using the New Pblock Command

The New Pblock command will simply create a new Pblock in the Physical Hierarchy view, but will not create a rectangle in the Device View. You should pre-select logic for assignment to the new Pblock. If no logic is pre-selected, an empty Pblock is created. To create new Pblocks with or without pre-selected logic, select **New Pblock** from the popup menu.

Note: Certain floorplanning advantages can be gained by creating Pblocks without rectangles. Logic assigned to the Pblock will receive an AREA_GROUP property for ISE with no RANGE defined. ISE uses the AREA_GROUP property to attempt to group the logic and to prevent it from being placed in other AREA_GROUP ranges.

Creating Multiple Pblocks with the Create Pblocks Command

Multiple Pblocks can be created in a semi-automated way by using the Create Pblocks wizard. The wizard will create a separate Pblock for each selected netlist instance. To use the wizard, pre-select a set of instances for inclusion into individual Pblocks.

To create multiple Pblocks for specific netlist instances:

1. Select the instances to include in a Pblock.
2. Select **Tools > Create Pblocks**.

The Create Pblocks wizard will appear. A list of the selected instances is displayed.

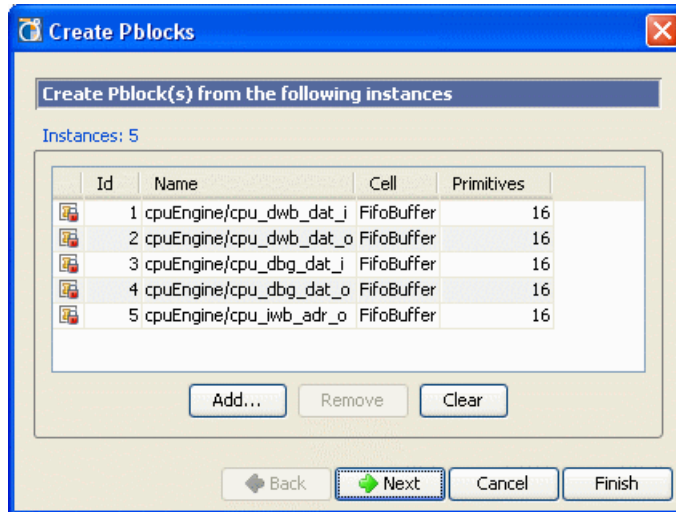


Figure 10-4: Create Pblocks Wizard: Create Pblocks from Instances

3. To add additional netlist instances to this list, click the **Add** button to invoke a browser in which you can select other instances.
4. To remove any netlist instances from the list, click the **Remove** button.
5. To clear netlist instances from the list, click the **Clear** button.
6. Click **Next**.

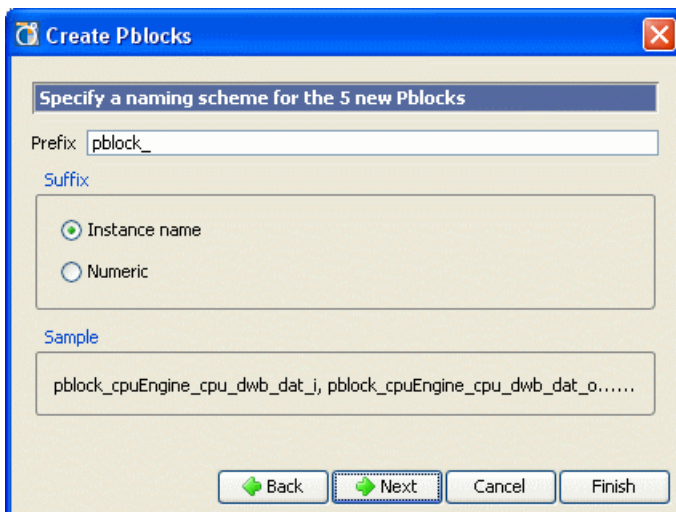


Figure 10-5: Create Pblock Wizard: Specify Name

7. In the Create Pblocks wizard, view and edit the naming scheme fields:
 - ♦ **Prefix**—Defines a name prefix to be used for the Pblock names. Enter a new prefix or allow the default instance name or number to be used.
 - ♦ **Suffix**—Select **Instance name** to append the instance name onto the prefix, or select **Numeric** to append a number starting with 1 to the prefix.
8. Click **Next**.
9. Verify the contents in the Summary page.
10. Click **Finish** to create the Pblocks with these settings.

The Pblocks should now be shown in the Physical Hierarchy view.

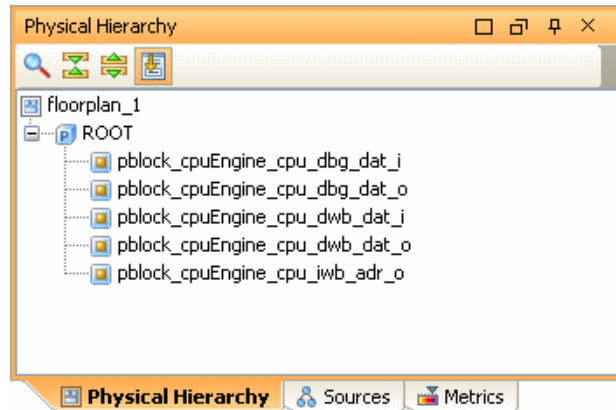


Figure 10-6: Pblocks in Physical Hierarchy

To create rectangles for the newly created Pblocks:

1. Select each of the new Pblocks (one at a time) in the Physical Hierarchy view.
2. Click the **Set Pblock Size** Device view toolbar button.



Figure 10-7: Set Pblock Size Toolbar Button

3. Draw a rectangle in the Device view.

Creating Non-Rectangular Pblocks

PlanAhead supports having non-rectangular Pblock shapes with multiple rectangles per Pblock. To add additional rectangles to existing Pblocks with rectangles, use the **Add Pblock Rectangle** toolbar button.



Figure 10-8: Add Pblock Rectangle

Pblocks with multiple rectangles will appear as separate rectangles with a dashed line connecting them.

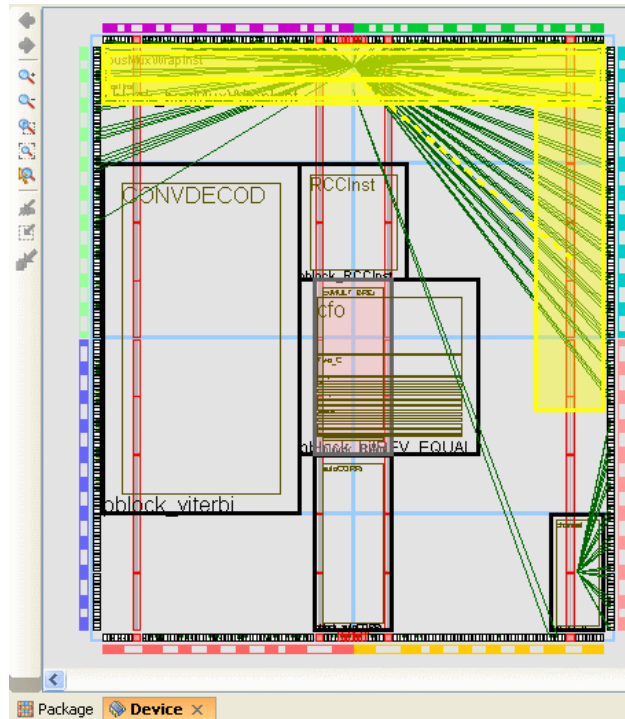


Figure 10-9: Pblock with Multiple Rectangles

Selecting a Pblock with multiple rectangles defined will select all of its rectangles. They may be moved individually, or together as a group.

To reshape one of the rectangles of a multiple rectangle Pblock, select a rectangle and use the Set Pblock Size command or manually stretch to resize it.

To select a rectangle individually, use one of the following methods:

- Select the **Select** popup command to select a single Pblock rectangle.
- Use the Pblock Properties Rectangles tab to select them individually.

Pblocks defined that span PowerPC and MGT sites may automatically receive multiple rectangle regions. This is done to enable the correct rectangle ranges to be defined for implementation.

Creating Nested Child Pblocks

Pblocks can be created within pblocks to provide further control for constraining logic. This can be extremely helpful when trying to improve performance of critical modules. The top-level pblock contain all the lower-level pblocks during utilization estimates.

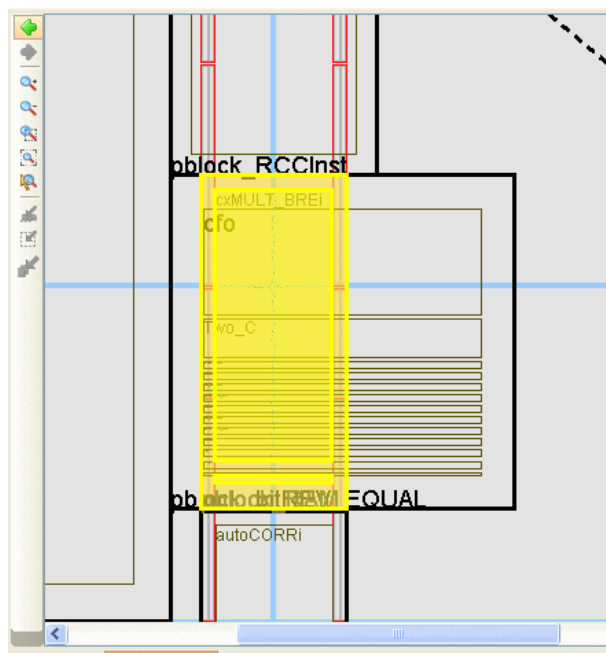


Figure 10-10: Creating Nested Pblocks

Note: The ISE implementation software does not support extensive use of this feature. Occasionally, map and placement errors will result when creating nested Pblocks.

Creating Clock Region Pblocks

A Pblock can be defined to include all resources within a specific clock region or regions. There is a set of steps required to define a Pblock as a clock region.

1. Draw a Pblock with a rectangle that encompasses the boundary of the clock region. PlanAhead displays the clock region boundaries. To change the color or display characteristics of the clock region boundaries, refer to the [“Customizing PlanAhead Display Options.”](#)

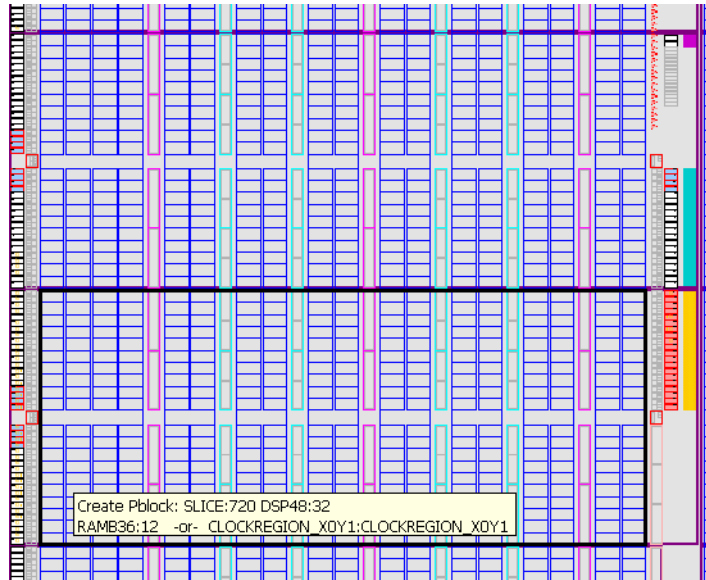


Figure 10-11: Creating Clock Region Pblocks

The tool tip changes to indicate the Pblock range is a Clock Region.

2. Select **OK** in the New Pblock dialog box to define the Pblock range as the Clock Region.

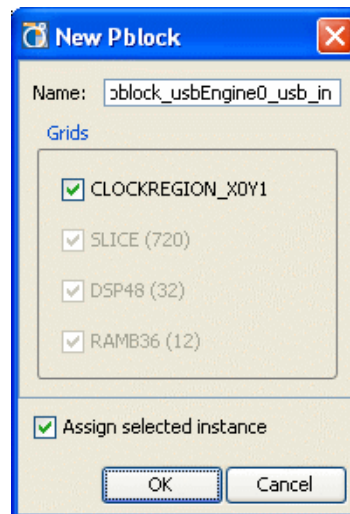


Figure 10-12: New Pblock Dialog to Confirm Pblock as Clock Region

The Pblock rectangle needs to encompass the clock region boundary to enable the CLOCKREGION option. Unselecting the CLOCKREGION_X button will enable the Pblock to be defined using traditional logic based ranges.

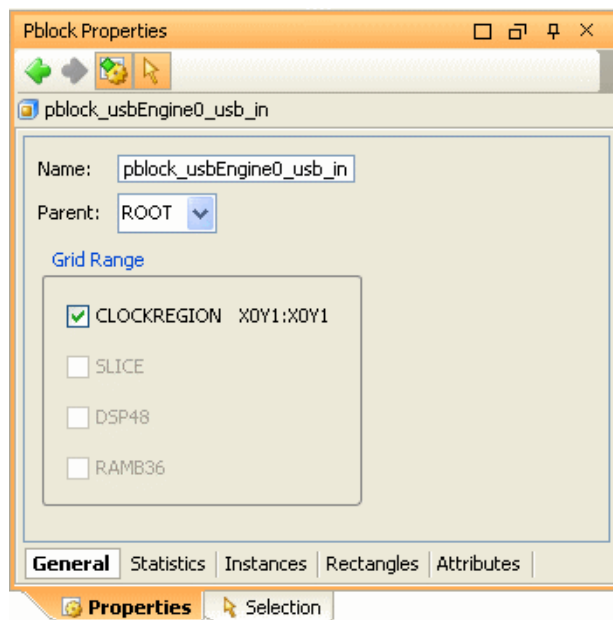


Figure 10-13: Clock Region Pblock General Properties

3. The two types of Pblocks can be toggled by selecting or unselecting the CLOCKREGION button in the New Pblock dialog or in the Pblock General Properties view.

The Pblock clock region coordinates are displayed in the Pblock General Properties view.

Working with Pblocks

Understanding Pblock Graphics

Using the default display options, Pblocks and the instances assigned to them are displayed. The outer rectangle is the Pblock border. The rectangles contained inside the Pblock are the netlist instances assigned to it. Multiple instances can be placed into a Pblock. Instance rectangles displayed inside the Pblock are sized based on the amount of logic they contain, relative to the other instances in the same Pblock. If many instances are assigned to a Pblock, they may appear as lines instead of rectangles.

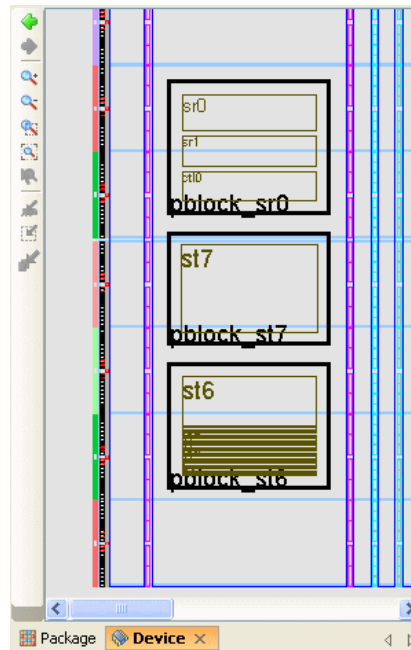


Figure 10-14: Pblocks With Assigned Instances Displayed Graphically

With the default selection rules set, selecting the Pblock rectangle will also select all of the netlist instances contained in it. Instances can easily be dragged and assigned into other Pblocks.

Note: Be careful when manipulating Pblocks to ensure the Pblock rectangle is selected and not the smaller rectangles indicating the instances assigned to it. It is often helpful when manipulating Pblocks to turn off the selection ability for instances. This ensures Pblocks and not the instances assigned to them are selected in the Device view. To define how instances and Pblocks are selected, select **Tools > Options > Themes > Device**, and define the selection ability in the Device dialog box. For more information, see “Setting Device View Display Options.”

I/O Nets are drawn connected to the center of the instance inside the Pblock rather than in the Pblock center as with Bundle Nets.

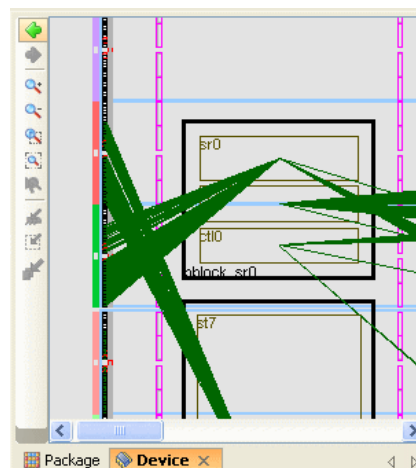


Figure 10-15: I/O Connectivity Displays to Center of Instance Rectangles

Child Pblocks appear in a slightly different color to differentiate the rectangles. Color configuration is available in the Tools > Options > Themes > Device dialog box.

Pblocks may contain multiple rectangle ranges. Multiple rectangle ranges are displayed with dashed lines connecting them to indicate that they are part of the same Pblock. The assigned instance rectangles and connectivity display in the largest rectangle.

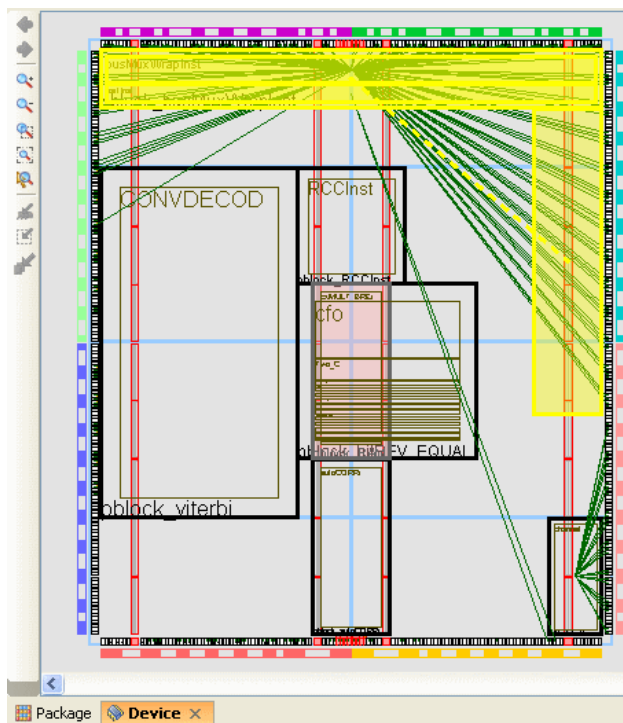


Figure 10-16: Creating Pblocks with Non-Rectangular Shapes

Assigning Logic to Pblocks

Once a Pblock has been created, netlist instances can be assigned to it. This can be done by either dragging and dropping logic, or by using the Assign popup command.

To use the drag and drop method:

1. Click and drag logic instances from the Netlist, Schematic, Hierarchy or Find Results views.
2. Drop them into the Pblock rectangle area.

To use the Assign command method to assign logic to existing Pblocks:

1. Select logic instances in the Netlist view.
2. Select the **Assign** popup command.

The Select Pblock dialog box will display the allowable selections of the Pblock assignment.

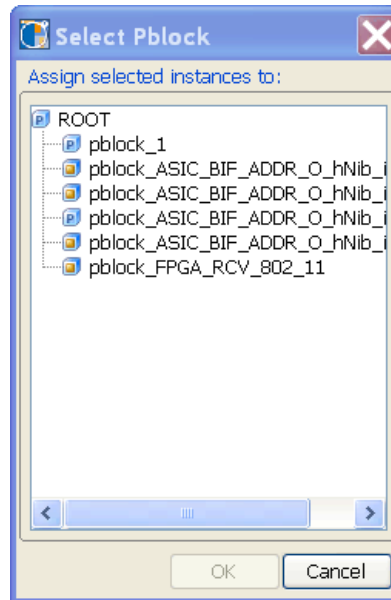


Figure 10-17: Select Pblock Dialog Box

Unassigning Logic from Pblocks

Instances may be removed from Pblocks, as follows:

1. Select the instances by any means.
2. Select the **Unassign** popup command.

A confirmation dialog box appears asking to confirm the removal of instances from the Pblock.

Moving a Pblock

Moving a Pblock is done by selecting and dragging the Pblock within the Floorplan, and dropping it in the new location. The dynamic cursor shaped like a hand indicates that the Pblock is selected for moving. Ensure that the outer Pblock rectangle is selected and not one of the instances assigned to it.

If the Pblock is moved to a location which includes new device logic types, such as BRAM, DSP, etc., a dialog box is displayed prompting you to add the new range types to the Pblock definition.

Pblocks behave different when location placement constraints are assigned inside of them. The desired target location should contain adequate resources to assign the placement constraints. As the Pblock is being dragged around, the cursor will indicate which are legal placement sites for a move. If not, a dialog box is displayed prompting you to either remove or leave the location constraints intact. Fixed and unfixed location constraints are listed separately in the dialog box, allowing you to handle them differently.

To cancel an active move operation, press the **Esc** key. The active command will be terminated.

Note: If having difficulty moving Pblocks, click the **Set Pblock Size** toolbar button to redraw the rectangle elsewhere. You may also wish to remove placement constraints prior to moving Pblocks.

Stretching a Pblock

Stretching Pblock edges may be done by selecting the Pblock and moving the cursor near one of its edges or corners. When the cursor changes to a drag symbol, click and drag to reshape the Pblock.

If the Pblock is stretched to a location which includes new device logic types, such as BRAM, DSP, etc., a dialog box is displayed prompting you to add the new range types to the Pblock definition.

Pblocks behave differently when location placement constraints are assigned inside of them. If location constraints are assigned to the Pblock, a dialog box is displayed prompting you to either remove or leave the location constraints intact. Fixed and unfixed location constraints are listed separately in the dialog box allowing you to handle them differently.

To cancel an active stretch operation, press the **Esc** key. The active command will be terminated.

Using the Set Pblock Size Command

Existing Pblocks can be sized or resized with a new rectangle by using the Set Pblock Size command. To create a new rectangle for an already existing Pblock:

1. Select the Pblock in the Physical Hierarchy view or Device view.
2. Click the **Set Pblock Size** toolbar button.



Figure 10-18: Set Pblock Size Toolbar Button

The cursor changes to enable you to draw a new rectangle in the desired location in the Device view.

3. Use the cursor to draw a new rectangle.

This command is also used to draw a rectangle for an existing Pblock with no rectangle yet defined, such as one created with the New Pblock(s) commands. For more information, see [“Creating Multiple Pblocks with the Create Pblocks Command.”](#)

If a Pblock has multiple rectangles, this command is used to regenerate the Pblock with a single rectangle. This is often useful when a Pblock gets fragmented into multiple rectangles.

If the Pblock is resized to a location which includes new device logic types, such as BRAM, DSP, etc., a dialog box is displayed prompting to add the new range types to the Pblock definition.

Pblocks behave differently when location placement constraints are assigned inside of them. If location constraints are assigned to the Pblock, a dialog box is displayed prompting you to either remove or leave the location constraints intact. Fixed and unfixed location constraints are listed separately in the dialog box allowing you to handle them differently.

To cancel an active resize operation, click the **Esc** key on the keyboard. The active command will be terminated.

Modifying Non-Rectangular Pblocks

Selecting a Pblock with multiple rectangles defined will select all of its rectangles. They may be moved individually, or together as a group.

To reshape one of the rectangles of a multiple rectangle Pblock, select a rectangle and use the Set Pblock Size command or manually stretch to resize it.

To select a rectangle individually, use one of the following methods:

- Select the **Select** popup command to select a single Pblock rectangle.
- Use the Pblock Properties Rectangles tab to select them individually.

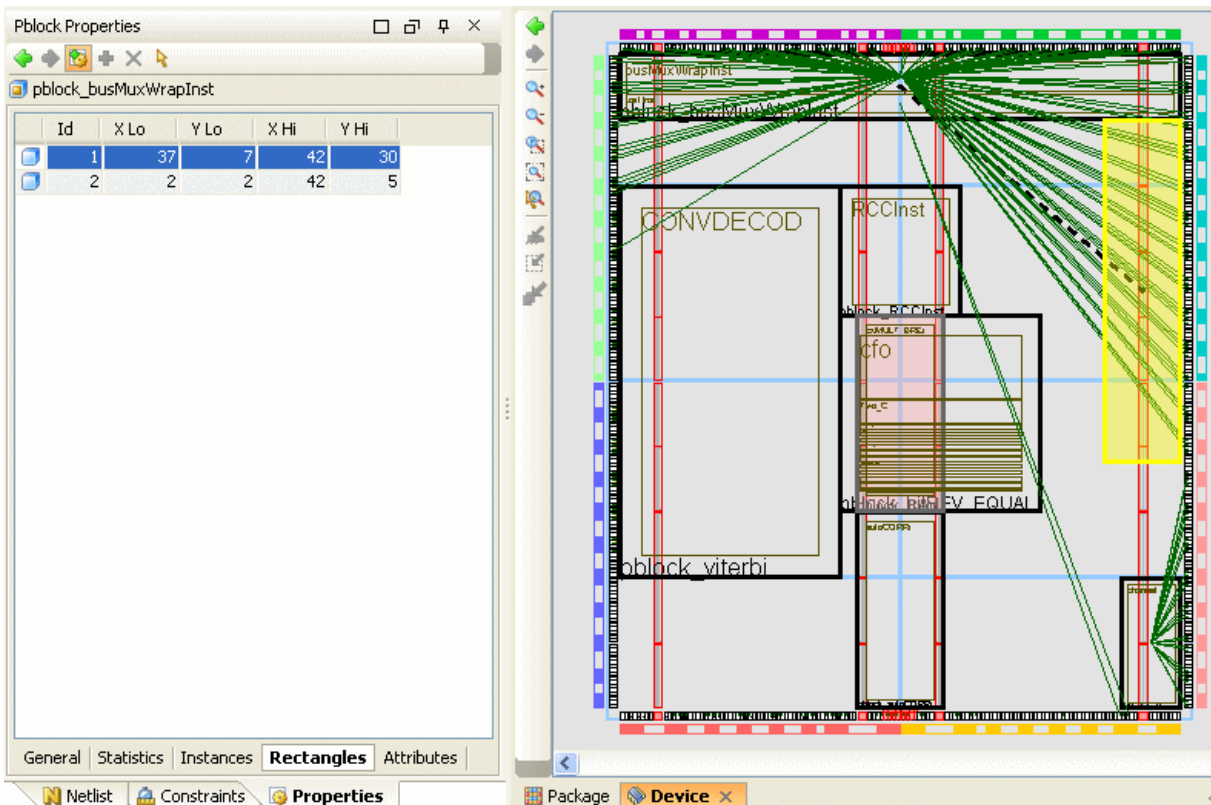


Figure 10-19: Selecting the Pblock Rectangles Individually

Pblocks defined that span PowerPC and MGT sites may automatically receive multiple rectangle regions. This is done to enable the correct rectangle ranges to be defined for implementation.

Note: The Xilinx® ISE implementation tools are not optimized to handle too many ranges per the AREA_GROUP constraint. It is best to use simple shape configurations such as L or T shapes.

Removing a Pblock Rectangle

The Pblock rectangle can be removed by selecting the Pblock and clicking the **Clear Pblock Rectangle** popup command.

Individual Pblock rectangles can be cleared one at a time. Multiple rectangles and Pblocks can be cleared simultaneously. Clearing Pblock rectangles does not delete the Pblock from the Physical Hierarchy.

Note: Certain floorplanning advantages can be gained by removing the Pblock rectangle and not the partition from the Physical Hierarchy view. The logic contained in the partition will receive an AREA_GROUP property for ISE. During placement, ISE will attempt to group the logic with AREA_GROUP properties and prevent it from being placed in other AREA_GROUP locations.

Renaming a Pblock

Pblocks can be renamed using the General tab of the Pblock Properties view. Enter the new Pblock name in the Name field and click **Apply**.

Deleting a Pblock

You can delete a selected Pblocks as follows:

1. Select one or more Pblocks in the Physical Hierarchy view.
2. Press the **Delete** key, or select the **Delete** popup menu command.
3. In the Confirm Delete dialog box, you can select the **Remove Pblock children** option to remove any nested Pblocks along with their partitions. Otherwise, when left unselected, you will delete the selected Pblock only and move any nested Pblocks up one layer of hierarchy.
4. Click **OK** to remove the Pblock partition from the Physical Hierarchy view.

Viewing or Changing Pblock Properties

Various types of information can be displayed with the Pblock Properties view. To display or edit Pblock properties, select the Pblock and view the Pblock Properties view. The tabs of this window are described below.

Note: To accept any changes made, click **Apply**. To cancel any changes, click **Cancel**. Selecting another item or closing the Properties view will not initiate any changes unless you click Apply.

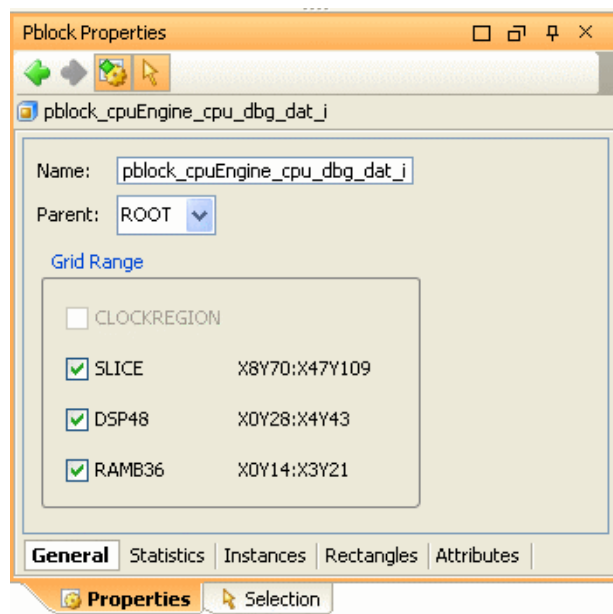


Figure 10-20: Pblock Properties View

The General Tab

The General tab contains the following definable fields:

- **Name**—Displays the Pblock name.
- **Parent**—Displays the Parent Pblock. This field is a non-editable field for some Pblocks. If a Pblock has multiple potential parent Pblocks, the field becomes active allowing definition of the Parent Pblock.
- **Grid Range**—Enables the Pblocks to be specified with specific AREA_GROUP RANGE properties. Selecting specific ranges will only constrain the selected logic types within the Pblock area. The grid range coordinates are displayed for each logic type once the Pblock is created.
 - ◆ **CLOCKREGION**—Select this button to define the Pblock range to be an entire clock region. The Pblock rectangle is drawn to match the clock region boundary.

The Statistics Tab

The Statistics tab of the Pblock Properties view displays content information about the Pblock. Note that you can also save the contents to a text file using the **Export Statistics** icon.

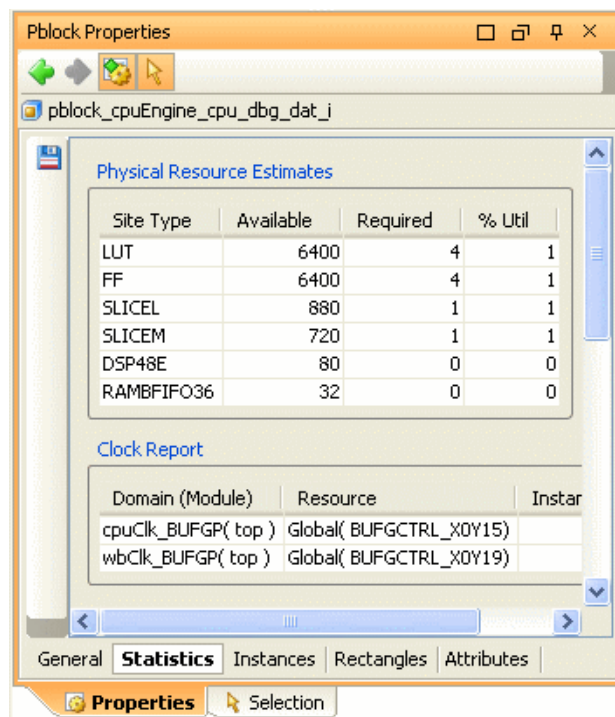


Figure 10-21: Pblock Properties View: Statistics Tab

The Statistics tab display the following Pblock information:

- **Physical Resources Estimates**—Displays a chart of each resource type in the device.
 - ◆ **Site Type**—Displays the Site Types defined within the Pblock rectangle.
 - ◆ **Available**—Displays the number of sites contained in the Pblock.

- ♦ **Required**—Displays the number of sites required for the logic assigned to the Pblock.
- ♦ **% Utilization**—Displays the estimated percentage of the sites populated in the Pblock.
- **Clock Report**—Displays all clock signals contained in the Pblock as well as the number of clocked instances on each clock. Local, Global, and Resource clocks are all displayed.
- **Carry Statistics**—Displays the number of vertical carry chain logic objects assigned to the Pblock. It also displays the tallest carry chain assigned to the Pblock and the percentage of its height in relation to the Pblock height. Carry height utilization values over 100% may cause PlanAhead DRC errors and ISE map errors.
- **RPM Statistics**—Displays the number of Relatively Placed Macros “RPM” objects assigned to the Pblock. It also displays the tallest and widest RPM assigned to the Pblock and the percentage of its size in relation to the Pblock size. RPM utilization values over 100% will cause PlanAhead DRC errors and ISE map errors. PlanAhead does not indicate whether multiple RPMs will fit inside the Pblock rectangle.
- **Clock Region Statistics**—Displays the utilization percentage of each clock region that the Pblock overlaps.
- **Net Statistics**—Displays the number of internal and external nets in the Pblock.
- **Cellview Statistics**—Displays the number of each type of logical resource assigned to the Pblock.
- **Boundary-crossing Nets Statistics**—Displays the number of nets that interface to the selected Pblock. Any net that is connected to logic inside of the Pblock is referred to as a boundary crossing net.

The Instances Tab

The Instances tab of the Pblock Properties view displays information about the instances contained in the Pblock.

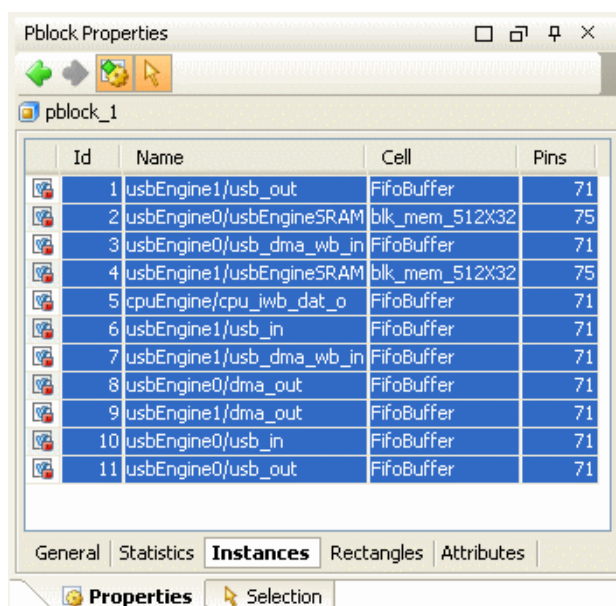


Figure 10-22: Pblock Properties View: Instances Tab

The instance fields are selectable and may be used to seed many popup menu commands.

The Rectangles tab

The Rectangles tab of the Pblock Properties view displays information about the various rectangles created for the Pblock. The Rectangle tab is used for selecting rectangles of a Pblock. For more information about this tab, refer to the [“Modifying Non-Rectangular Pblocks.”](#)

The Attributes tab

The Attributes tab of the Pblock Properties view is described in the next section.

Setting Pblock Logic Types Ranges

You can set Pblock AREA_GROUP Range types in the Pblock Properties view by modifying the Grid Range options in the General tab. Adjusting these toggles will control which type of logic is to be constrained within the Pblock rectangle.

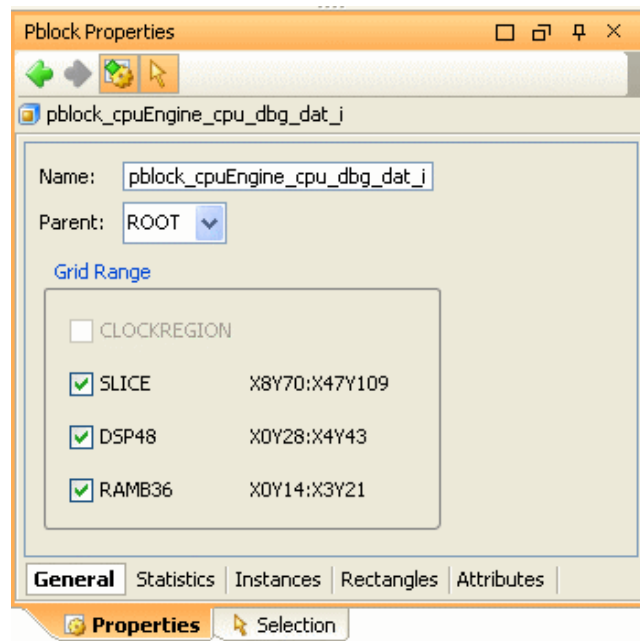


Figure 10-23: Setting AREA_GROUP Range Types

If the Pblock is resized or moved to a location that includes new device logic types, such as BRAM, DSP, etc., a dialog box is displayed prompting to add the new range types to the Pblock definition.

Toggleing the ranges *off* will result in the Pblock being shown differently in the Device view. As the Pblock is selected the shading will only affect the logic types for the ranges set on the Pblock, as shown below.

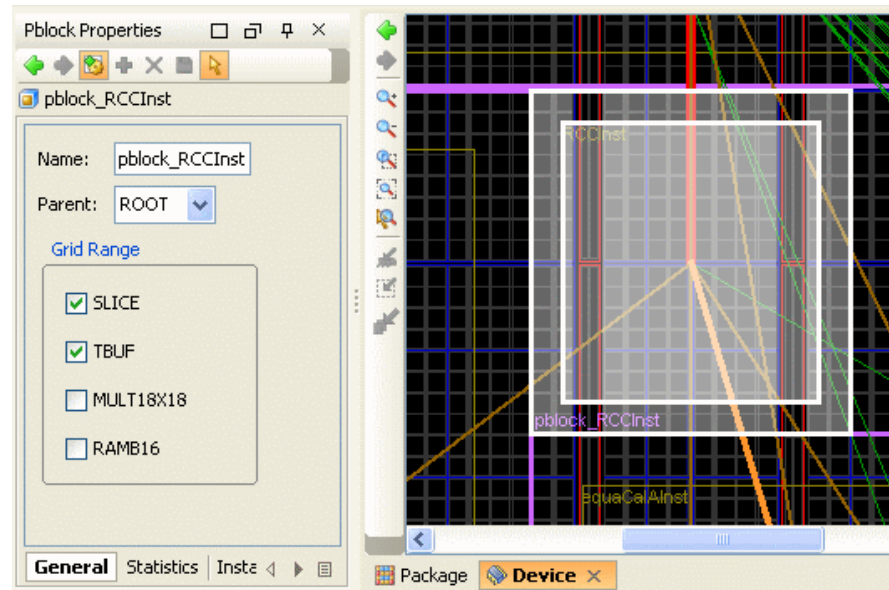


Figure 10-24: Pblock Shading Reflects Logic Contained in Pblock

Setting Attributes for Pblocks

Attribute properties can be assigned to Pblocks in the Pblock Properties Attributes tab. Assigning attributes set various options for ISE.

Note: Setting these attributes can affect implementation results or cause failures.

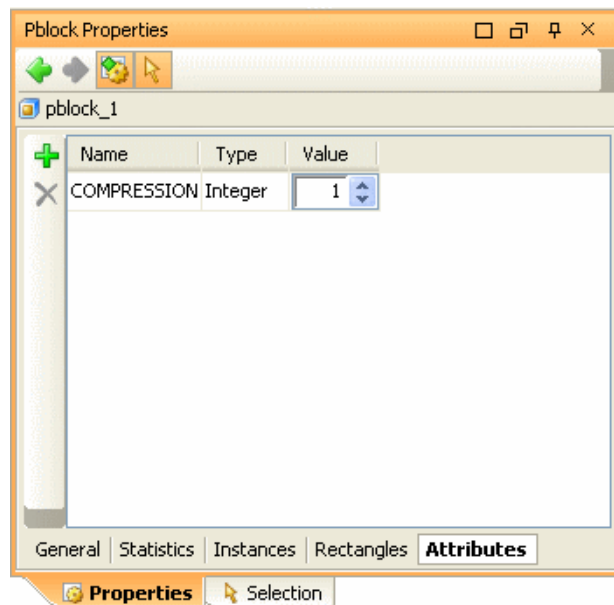


Figure 10-25: Pblock Properties View: Attributes Tab

To define new attributes for the Pblock, in the Attributes tab of the Pblock Properties View:

1. Select **Define** from the popup menu, or click the **Define new attribute** toolbar button.



Figure 10-26: Define New Attribute Toolbar Button

The Define Attribute dialog box displays.

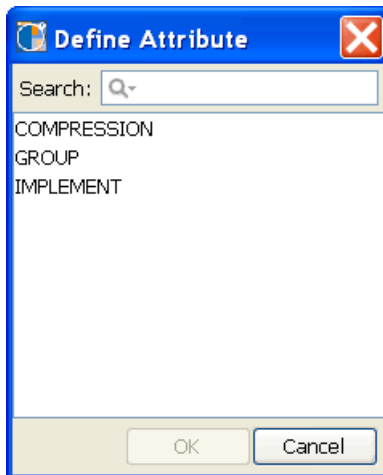


Figure 10-27: Define Attribute Dialog Box

2. Select the desired attribute to assign.
3. Click **OK**.

The specified attribute type is added in Attributes tab.

4. You can then specify an attribute value.

The available Pblock Attributes are as follows:

- ♦ **COMPRESSION**—Controls the compression factor for the area group. The percent values can be from 0 to 100, with 0 being no compression, and 1 being maximum compression. COMPRESSION will not work with Virtex®-5 or Virtex-6 devices, and any design using **map -timing**.
 - ♦ **GROUP**—Controls the packing of logic into physical components (that is, slices), allowing logic outside the area group to be with logic inside the area group.
 - ♦ **IMPLEMENT**—Controls whether the area group logic will be reimplemented.
5. Click **Apply** to accept the changes.

Using Resource Utilization Statistics to Size Pblocks

Pblocks can be sized and placed using the utilization estimates in the Pblock Properties view. The device resources available inside the rectangle are compared against the logic contained in the Pblock to compute utilization estimates.

To display the utilization estimates for a Pblock:

1. Select the Pblock, and view the Pblock Properties.
2. Click the **Statistics** tab.

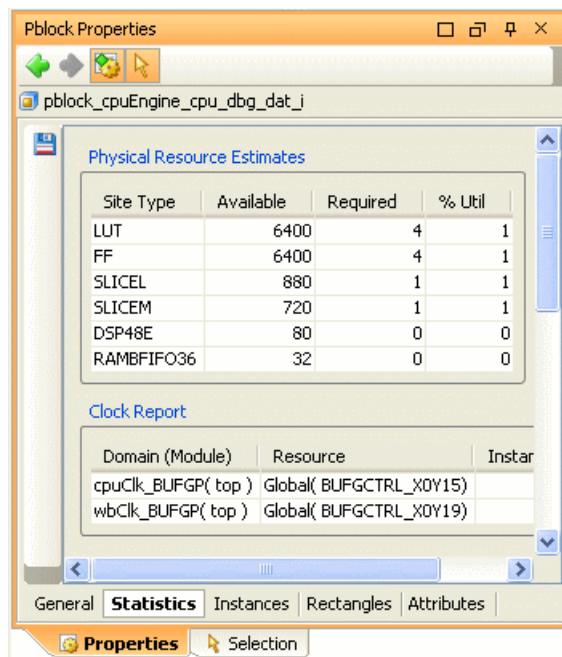


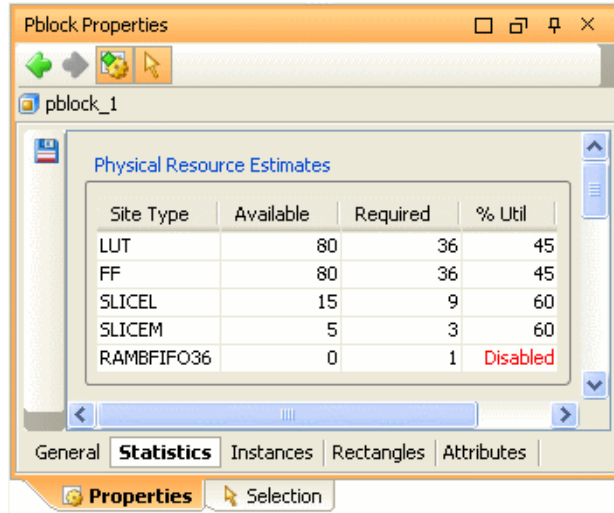
Figure 10-28: Pblock Properties View: Statistics Tab

3. In the Statistics tab, view the utilization estimates in the following three columns.
 - ♦ Available—Displays the number of available sites in the Pblock.
 - ♦ Required—Displays the number of sites that the assigned logic requires.
 - ♦ % Utilization—Displays the estimated utilization percentage for each logic type. The appropriate utilization can be met by resizing the Pblock.
4. Scroll down to view the required RAM sites for the Pblock.

The dialog box is dynamic and will update each time the Pblock is modified.

If the Pblock does not contain a site for a specific logic device element, the following values are shown:

- Available—0
- Required—the number required
- Utilization—Disabled (meaning no sites of the required type are defined in the rectangle). A value of *Disabled* is an error condition indicating that there are no sites of the required type in the rectangle defined.



Pblock Properties

pblock_1

Physical Resource Estimates

Site Type	Available	Required	% Util
LUT	80	36	45
FF	80	36	45
SLICEL	15	9	60
SLICEM	5	3	60
RAMBFIFO36	0	1	Disabled

General **Statistics** Instances Rectangles Attributes

Properties Selection

Figure 10-29: No RAMBFIFO36 Resources Available in Pblock (Disabled)

Note: The Pblock SLICE utilization calculation assumes maximum site utilization. In reality, the maximum site utilization is rarely achieved in placement and routing tools. Thus, a designer should optimize for a target utilization of approximately 80% or higher. This number is a function of the device used and the characteristics of the design and its constraints.

Note: Pblock utilization is affected by carry chains, RPM macros and the geometry of the Pblock rectangle. These statistics are merely estimates to help guide you to a successful ISE implementation. All of the Pblock Statistics should be taken into account when sizing Pblocks. Occasionally, Pblocks will need to be enlarged in order for ISE to place them successfully.

Placing Pblocks Based on Connectivity

PlanAhead provides dynamic connectivity feedback to help guide placement of Pblocks.

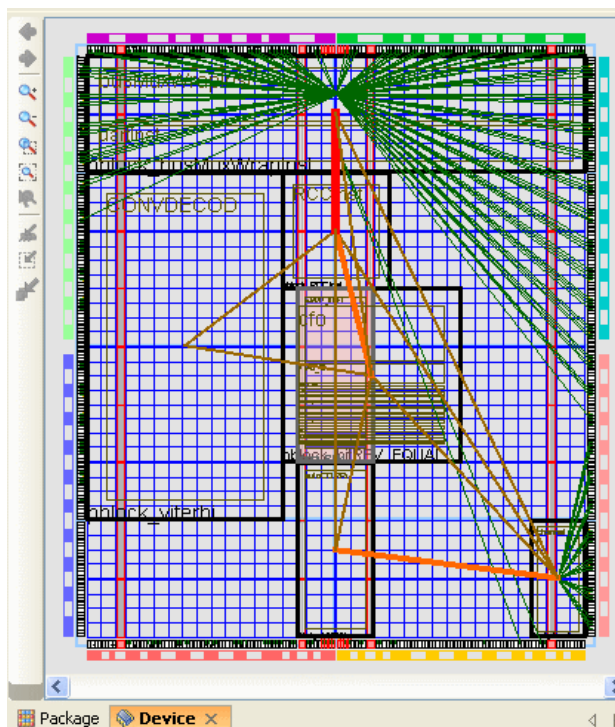


Figure 10-30: **Displaying Connectivity Helps To Determine Floorplan**

The combined connectivity between Pblocks is displayed as bundled nets. Each bundle is sized and colored based on the number of connections between Pblocks. The intent is to make the heavily connected Pblocks easy to identify. A reasonable strategy might be to define the Pblocks with the largest net bundles close together. Typically, the Pblocks should be placed in such a way as to achieve the shortest net lengths and to avoid routing conflicts or congestion.

Displaying Bundle Net Properties

Connectivity information can be viewed by displaying properties for net bundles or for individual nets. To view connectivity information:

1. Select the desired net or bundle net.
2. View the Net Properties or Bundle Net Properties view.

The Nets tab in the Bundle Net Properties dialog box displays the nets contained in the bundle.

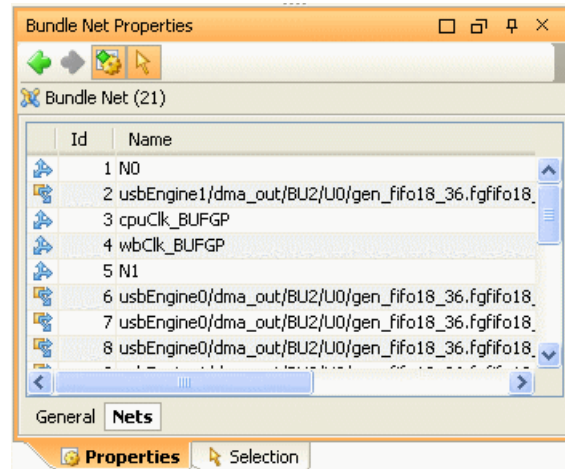


Figure 10-31: Bundle Net Properties: Nets Tab

Adjusting Bundle Net Defaults

The color, line width and signal count range can be defined in View-specific settings in the Themes dialog box (**Tools > Options > Themes**). For more information on setting Bundle Net defaults, see [“Setting the Device View Bundle Nets Display Options.”](#)

Using the Automatic Pblock Commands

PlanAhead enables automatic partitioning and Pblock placement using the Auto-create Pblocks command. This method is used primarily to create a top-level Floorplan to view the data flow of the design and to understand the relative size and relationship between the various logic modules in the design. Normally, the designer will have some concept of the critical modules and circuitry in the design and begin floorplanning with those modules.

The automatic floorplanning features are not meant to be used as the only floorplanning methodology. They are to be used as a tool to help the designer understand the physical design. Floorplanning is a manual process that leverages designer insight to help guide the ISE tools. The automatic placement features available in PlanAhead are not a shortcut to floorplanning the design. You may still need to make partitioning decisions and shape Pblocks accordingly to ensure that non-SLICE based logic is accounted for in the Pblock areas.

Automatically Creating Pblocks

The Auto-create Pblocks command is an automatic partitioning tool that can operate on either the top level of the netlist or at any logical instance level in the netlist. By default, the Auto-create Pblocks command starts at the top-level, and works down the hierarchy one module at a time. It can create a Pblock for each logic module in the underlying netlist or it can be filtered to only create a subset.

This command will automatically place the instances into Pblocks and name them accordingly.

To run the Auto-create Pblocks command:

1. Select **Tools > Auto-create Pblocks**.

The Auto-create Pblocks dialog box appears.

Note: Pblocks can be preselected to seed the Auto-create Pblocks dialog box.

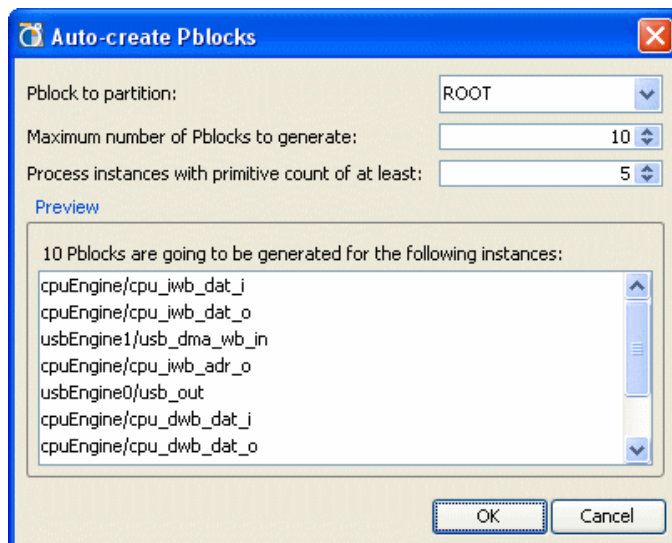


Figure 10-32: Auto-create Pblocks Dialog Box

2. View and edit the definable options in the Auto-create Pblocks dialog box:
 - ♦ **Pblock to partition**—Select the partition level. It can either be the top-level Pblock, “ROOT”, or any other pre-selected Pblocks.
Note: PlanAhead partitions are not related to ISE partitions. The PlanAhead partition command does not create ISE partitions.
 - ♦ **Maximum number of Pblocks to generate**—This instructs the Auto-create Pblocks command to create no more Pblocks than the number set. If the number of modules exceeds the number set, the largest <number set> of Pblocks will be created.
 - ♦ **Process instances with instance count of at least**— This instructs the command to not create Pblocks with a number of instances less than the number set.
 - ♦ **Preview**—Displays the Pblocks that will be created.
3. Click **OK** to automatically create the Pblocks in the design.

Pblocks are automatically created and named accordingly for each of the top-level Netlist modules, as shown in the following figure. No rectangles are created yet.

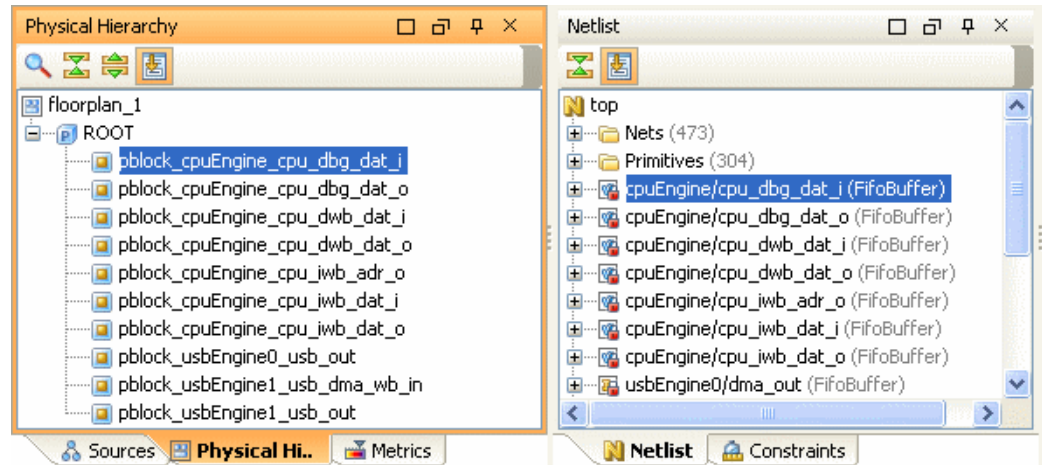


Figure 10-33: New Auto-Created Pblock

Running the Automatic Pblock Placer

The PlanAhead Pblock placer will automatically size and place pblocks in the device. Pblocks are sized based on SLICE content only. All other device types are ignored when Pblocks are created. The Pblocks are created with all RANGES defined.

The Pblock Placer command is intended to provide a quick placement of the selected pblocks. This is often very helpful to view the data flow through the design. You are required to manually adjust Pblock shapes before ISE implementation to include non-SLICE resources.

Note: The resulting Pblocks from the Place Pblocks command may not be suitable for ISE implementation. They are sized based on SLICE logic only. You are required to manually size the Pblocks to include non-SLICE based logic resources.

Note: The Automatic Pblock Placer is not available for Virtex-6 or Spartan®-6 devices.

To run the block placer:

1. Select **Tools > Place Pblocks**.

The Place Pblocks dialog box appears.

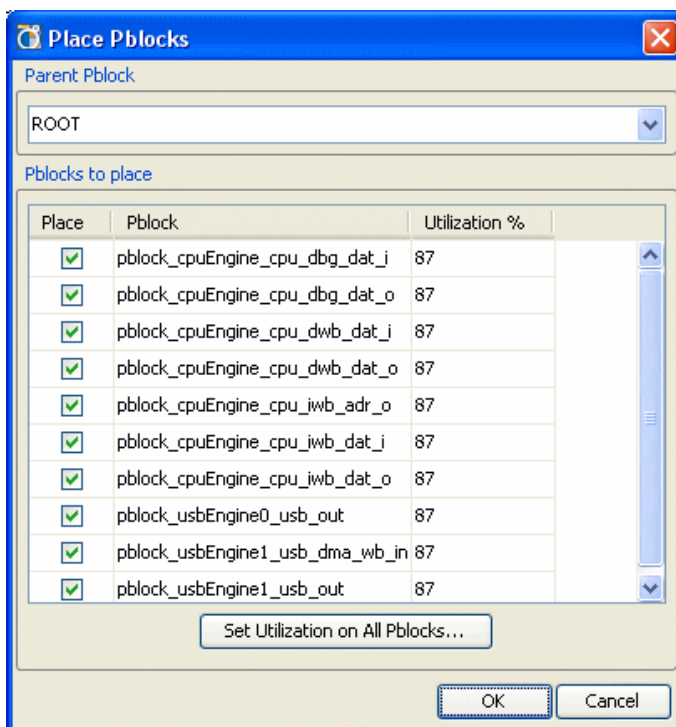


Figure 10-34: Place Pblocks Dialog Box

2. View and edit the definable options in the Place Pblocks dialog box:
 - ♦ **Parent Pblock**—Select the level of hierarchy to place Pblocks. Pblocks can be placed at the top “ROOT” or any partitioned Pblock level of hierarchy.
 - ♦ **Pblocks to place**—The Pblocks that exist under the parent Pblock are displayed.
 - **Place**—The check boxes control the Pblocks to be placed. Deselecting Place preserve existing Pblock rectangle locations.
 - **Pblock**—Lists all Pblocks.
 - **Utilization**—Enables specific SLICE utilization targets to be set for each Pblock.
 - ♦ **Set Utilization on all Pblocks**—Specifies a new target SLICE utilization target for Pblocks.
3. Click **OK** to place Pblocks in the design.

A Place Pblocks progress meter is displayed while the Place Pblocks command is running. The Pblocks will be automatically sized and placed based on SLICE utilization only.

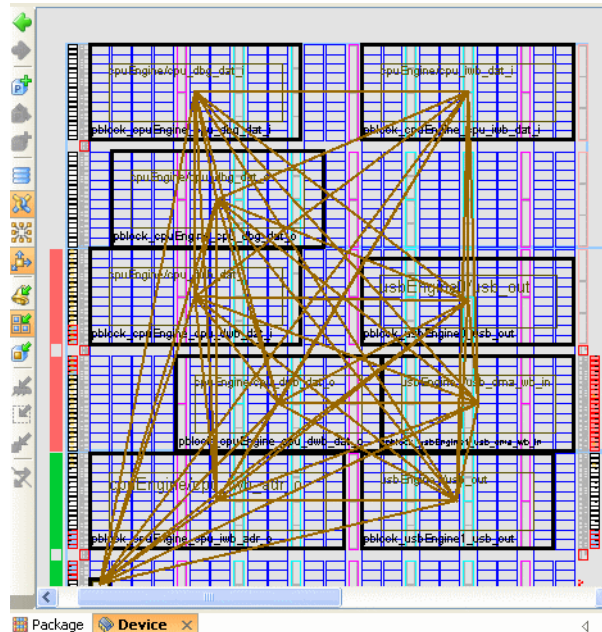


Figure 10-35: Auto-Placed Pblocks

Larger Pblocks can be partitioned again and Place Pblocks command can be performed within them.

Refer to the PlanAhead Tutorials for ways to create top-level Floorplans using the Partition and Place Pblocks commands.

Working with Placement LOC Constraints

Primitive logic elements can be assigned to specific logic sites using either the Create Site Constraint Mode or Create BEL Constraint Mode. PlanAhead includes BEL-level constraint assignments to lock logic to specific gates within the site.

Understanding Fixed and Unfixed Placement Constraints

PlanAhead differentiates placement constraints that are user assigned to those that are assigned by the ISE implementation tools. User assigned constraints are either defined within imported UCF files or manually assigned in PlanAhead. Constraints that are user assigned are considered “fixed” and are displayed in a different color. All placement constraints imported from ISE are considered “unfixed”. Placement constraints can be selected and set to fixed using the **Fix Instances** popup menu command.

All fixed constraints are exported by default to the ISE implementation tools in order to lock the placement. If you wish to lock down unfixed placement constraints, export the unfixed placement constraints from the Run Launch Options dialog box, the Export Constraints dialog box, or the Export Pblock dialog box.

Understanding Site and BEL Level Constraints

Site constraints result in a LOC constraint being assigned to the instance in the saved and exported Floorplan UCF files. The logic element is locked to the CLB SLICE site only, and not to any specific gate within it.

```
INST "receiver/uartInst/G_98_1" LOC = SLICE_X49Y69;
```

BEL constraints result in a LOC constraint and a BEL constraint being assigned to the instance in the saved and exported Floorplan UCF files. BEL constraints will assign a logic element to a specific gate within the CLB.

```
INST "channel/receiverRE[8]" BEL = FFX;  
INST "channel/receiverRE[8]" LOC = SLICE_X59Y2;
```

Assigning Site Location Placement Constraints (LOCs)

A leaf-level primitive instance can be placed into a specific device resource site by dragging it from the netlist tree and dropping it onto a specific site. Placing instances into sites adds Instance location constraints (LOCs) to the exported UCF files for ISE. The assigned locations will be assigned as fixed and locked during subsequent ISE attempts.

Before assigning instances, click the **Create Site Constraint Mode** toolbar button to initiate the Create Site constraint mode.



Figure 10-36: **Create Site Constraint Mode Toolbar Button**

The dynamic cursor will not allow instance placement to an illegal site or one that is already occupied. A legal placement site is indicated when the dynamic cursor changes from a slashed circle to an arrow or diamond. The dynamic cursor will not allow instances to be placed if the SLICE will be over packed with logic. Certain logic groups, such as carry-chain logic, move as a single object which requires open placement sites for all logic on the carry chain.

After location constraint assignment is complete, return to the default Assign instance to Pblock mode by clicking the **Assign Instance Mode** toolbar button.



Figure 10-37: **Assign Instance Mode Toolbar Button**

To view location constraint properties, select the placement constraint, and view the Instance Property view.

Assigning BEL Placement Constraints (BELs)

A leaf-level primitive instance can be placed into a specific device gate site by dragging it from the netlist tree and dropping it onto a specific site. Placing instances into gates assigns BEL constraints for ISE, as described above. The assigned locations will be assigned as fixed and locked during subsequent ISE attempts.

Before assigning instances, toggle the **Create BEL Constraint Mode** toolbar button on to initiate the Create BEL constraint mode.



Figure 10-38: **Create BEL Constraint Mode Toolbar Button**

The dynamic cursor will not allow instance placement to an illegal or already occupied gate sites. A legal placement site is indicated when the dynamic cursor changes from a slashed circle to an arrow. The dynamic cursor does not allow instances to be placed if the SLICE will be over packed with logic.

After location constraint assignment is complete, return to the default Assign instance to Pblock mode by clicking the **Assign Instance Mode** toolbar button.



Figure 10-39: **Assign Instance Mode Toolbar Button**

To view location constraint properties, select the placement constraint, and view the Instance Property view.

Adjusting the Visibility of Placement Constraints

Adjust the zoom level to change how assigned placement constraints are displayed. From a zoomed out view, the LOCs and BELs are displayed as a filled in rectangle inside the assigned site. When the zoom level increases, the logic is displayed as being assigned to specific logic gates within the site.

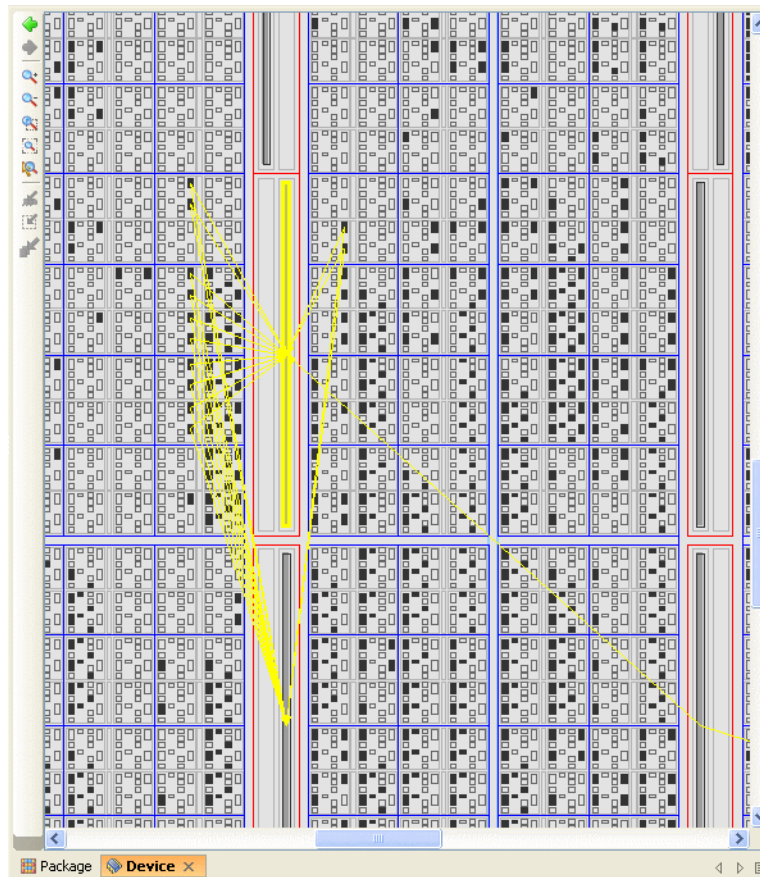


Figure 10-40: **Zoomed Out Device View**

To display or hide the location constraints, click the **Show/Hide LOC Constraints** toolbar button.



Figure 10-41: **Show/Hide LOC Constraints Toolbar Button**

To adjust other display characteristics for the LOC and BEL constraints:

1. Select **Tools > Options > Themes > Device**.

The Device dialog box displays.

2. Adjust values under the Constraint object.

To adjust the display of the device resource grid sites, select **Tools > Options > Themes > Device**, and in the Device dialog box, toggle values under the Site object.

Fixed and unfixed placement constraints have individual color and selection controls.

Viewing Logic Connectivity using Show Connectivity Mode

When placement constraints are visible, net connectivity is displayed to the individual placement constraint locations. When placement constraints are not visible, net connectivity is displayed to the center of the logic instance within the Pblock.

To display connectivity for a selected logic instance or group of instances, select one of the following commands:

- The **Show Connectivity** popup command.
- The **Show connections for selected instances Mode** toolbar button.



Figure 10-42: Show connections for selected instances Mode Toolbar Button

The *Show connections for selected instances mode* will remain active allowing additional logic objects to be selected for viewing connectivity. Toggle the toolbar button to turn off *Show connections for selected instances mode*.

Moving Placement Constraints

To move placement constraints:

1. Select a placement constraint by clicking on the Instance in the Device view, Netlist view or Schematic view.

2. Drag and drop the selected placement constraint to another legal site.

The primitive instance is assigned to the new site. Net flight lines can be displayed from the location constraint to connected Pblocks.

Moving a logic object that is part of a carry chain will result in the entire carry chain being selected for move. The cursor will indicate legal placement sites for the entire carry chain and all objects will move to new relative locations.

Logic such as RAMs and MULTs can be assigned to sites outside of the Pblock rectangle. This allows flexibility when locking placement for these types of logic elements.

3. After location constraint assignment is complete, return to the default Assign instance to Pblock mode by clicking the **Assign Instance Mode** toolbar button.



Figure 10-43: Assign Instance Mode Toolbar Button

To view location constraint properties, select the placement constraint, and view the Instance Property view.

Deleting Selected Placement Constraints

Selected instance location constraints can be deleted by selecting the placed instances, and using one of the following methods:

- Select **Tools > Clear Placement Constraints**. For more information, see [“Selectively Clearing Placement Constraints.”](#)
- Click the **Unplace** popup menu command.

Selectively Clearing Placement Constraints

Instance location constraints can be selectively removed from the design. You can filter the type of constraints you want to clear based on ISE assigned, selected logic or Pblocks and specific logic types.

Preselected objects will dictate the behavior of the Clear Placement Constraints wizard. If Pblocks are pre-selected, the wizard will display default settings to clear placement constraints inside of them. If Instances are selected, the wizard will display default settings to remove them.

To clear placement constraint assignments:

1. Select **Tools > Clear Placement Constraints**.

The Clear Placement Constraints wizard is invoked.

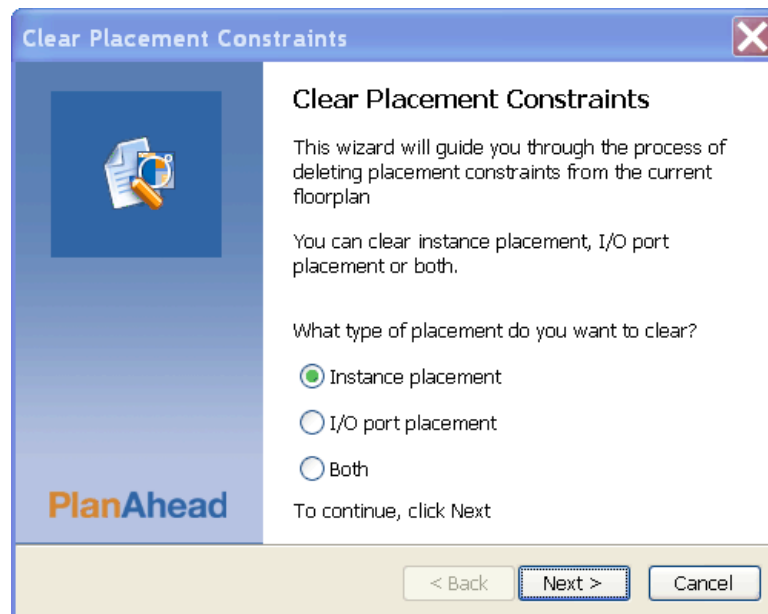


Figure 10-43: Clear Placement Constraints Wizard: Clear Instance Placement, I/O Ports Placement or Both

2. In the Clear Placement Constraints wizard, specify the type of placement constraints you wish to remove: Instance placement, I/O Port placement, or both.
3. Click **Next**.

Instance placement

The next dialog will differ depending on what types of objects were selected prior to invoking the command. If nothing was selected, it is not shown. If a Pblock is selected, it defaults to clear the contents of the Pblock. If instances are selected, it defaults to remove them. Additional options are presented depending on the preselected set, as shown below.

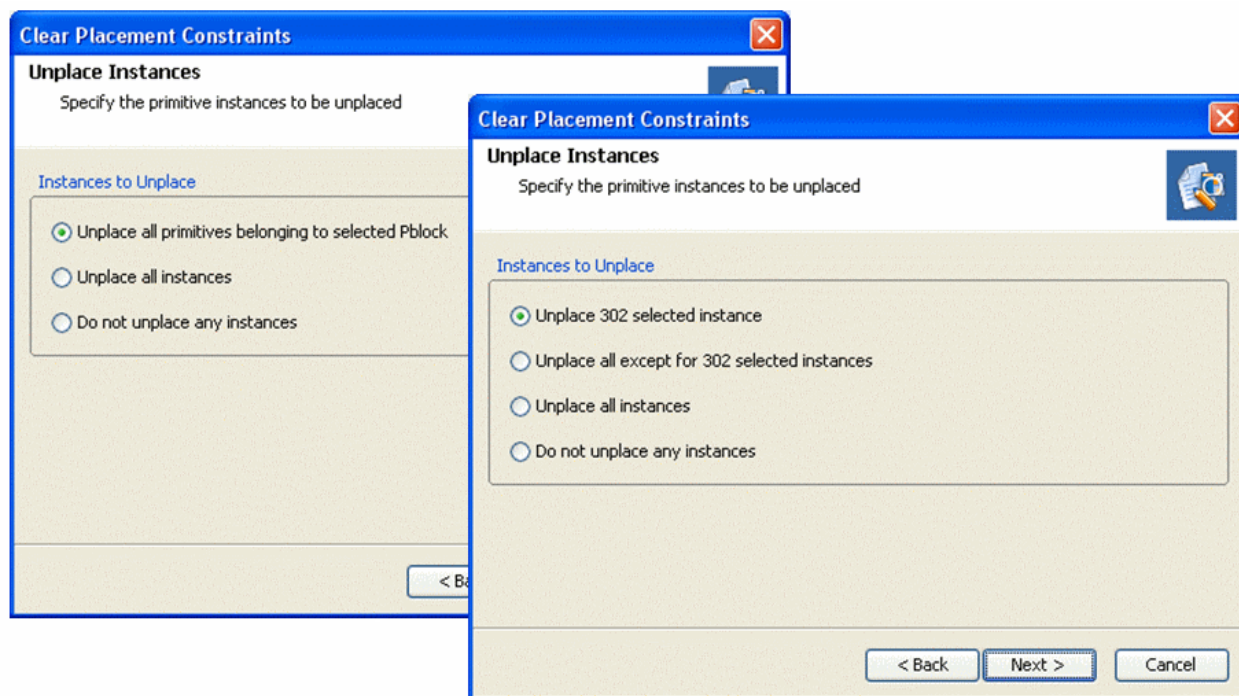


Figure 10-44: Clear Placement Constraints Wizard: Placement Removal Options Based on Pre-Selected Objects

4. In the Unplace Instances page, select the category of instances to be unplaced.
5. Click **Next**.

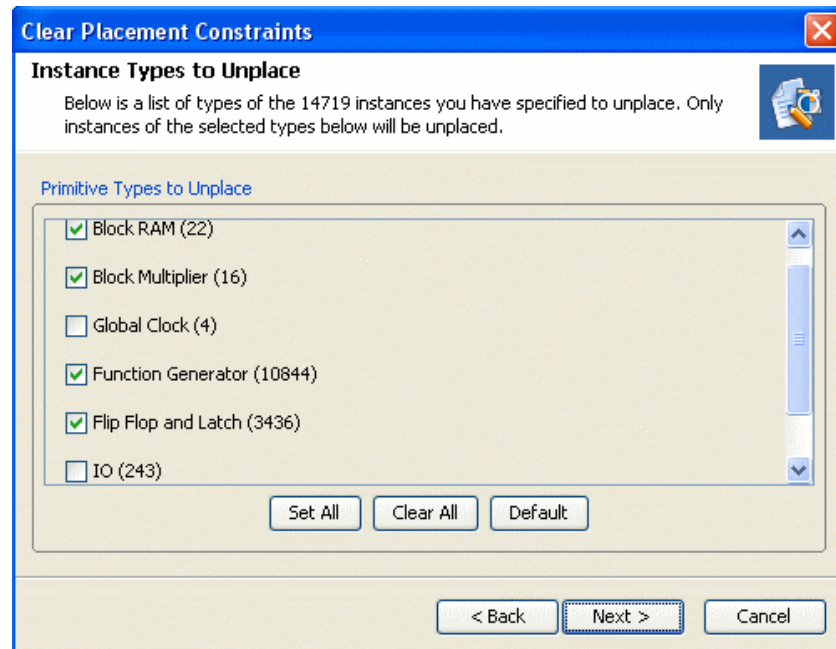


Figure 10-45: Clear Placement Constraints Wizard: Filter Logic Types to Remove

The Instance Types to Unplace page provides a mechanism to filter the types of placement constraints to filter. The I/O related boxes are unchecked because we elected to clear instance constraints in the first page of the wizard only.

6. In the Instance Types to Unplace page, select the Primitive Types checkbox.
7. Click **Next**.

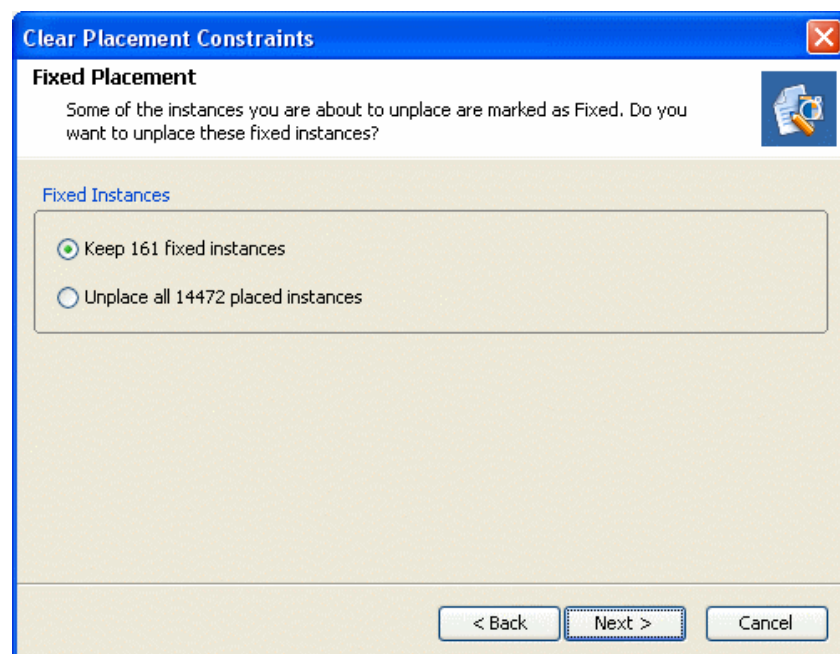


Figure 10-46: Clear Placement Constraints Wizard: Filter Fixed Constraints

8. In the Fixed Placement page, specify whether or not to unplace the fixed instance(s). Fixed instances are those you placed or “fixed” in your design, or those you imported in the input UCF files.
9. Click **Next**.
10. Verify the contents of the Summary page, and click **Finish**.

The specified primitive instance assignments are removed from the design.

I/O Port placement

To clear I/O Port constraint assignments:

1. Select **Tools > Clear Placement Constraints**.

The Clear Placement Constraints wizard is invoked.

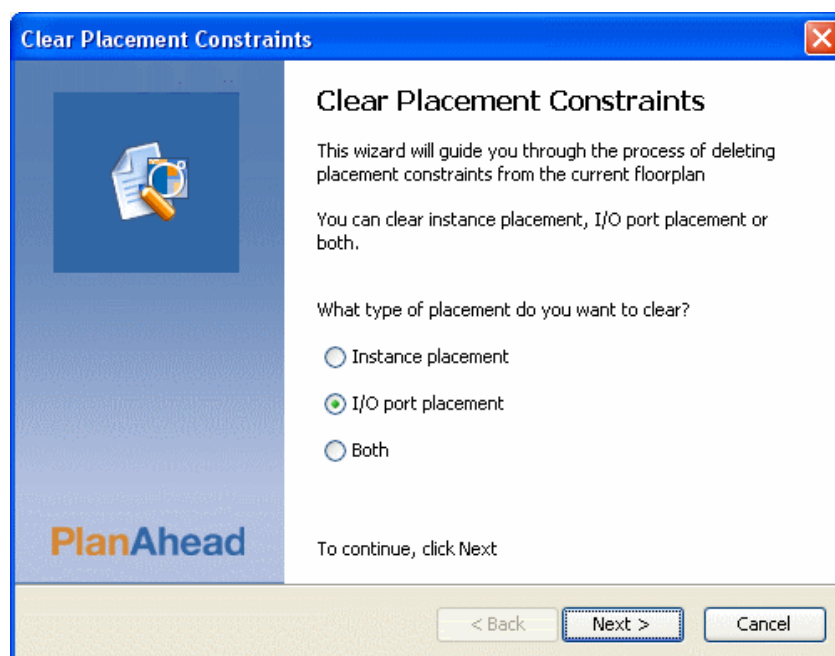


Figure 10-47: **Clear Placement Constraints Wizard: Clear I/O Ports or Both**

2. In the Clear Placement Constraints wizard, specify the type of placement to clear: Instances placement, I/O Port placement or Both.
3. Click **Next**.

The next dialog box will differ depending on what types of objects were selected prior to invoking the command. If nothing was selected, it is not shown. If I/O Ports are selected, it defaults to remove them. Additional options are presented depending on the preselected set, as shown below.

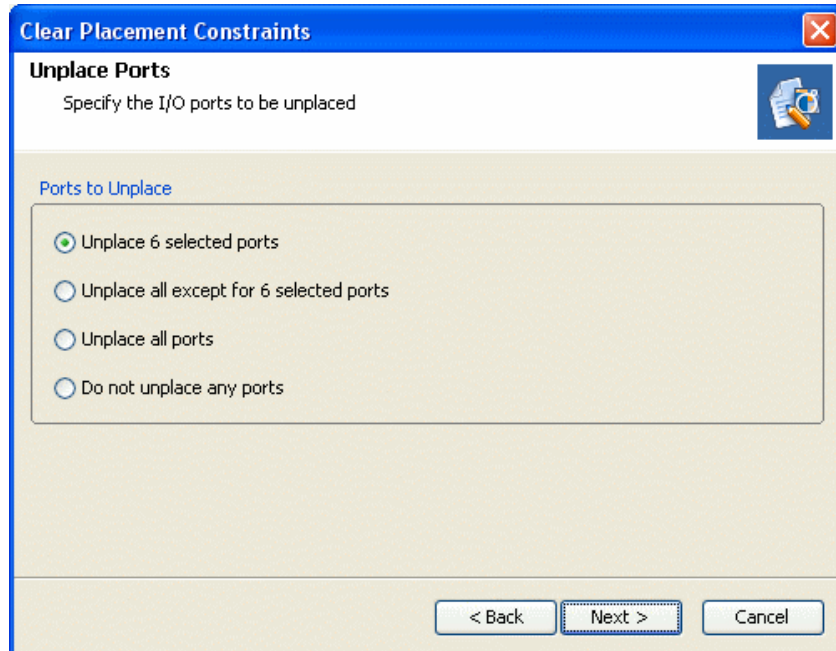


Figure 10-48: Clear Placement Constraints Wizard: I/O Port Removal Options Based on Pre-Selected Objects

4. In the Unplace Ports page, select the category of I/O ports to be unplaced.
5. Click **Next**.

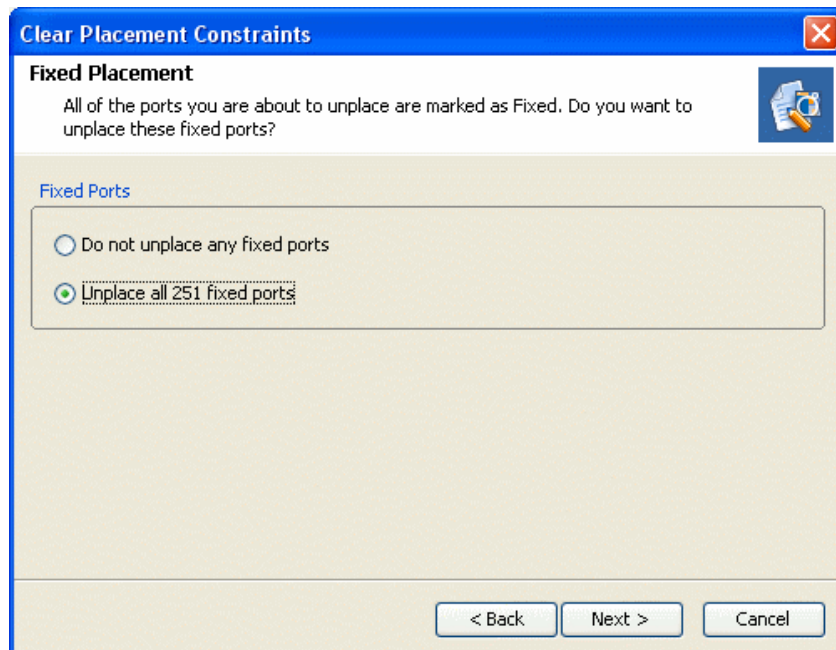


Figure 10-49: Clear Placement Constraints Wizard: Filter Fixed Constraints

6. In the Fixed Placement page, specify whether or not to unplace the fixed instance(s). Fixed instances are those you placed or “fixed” in your design, or those imported in the input UCF files.

7. Click **Next**.
8. Verify the contents of the Summary page, and click **Finish**.

The specified I/O ports assignments are removed from the design.

Previously assigned ports are not cleared prior to reading a new UCF file. New port assignments write over previous assignments. It is best to first clear all port assignments prior to importing new port assignment constraints.

Moving Pblocks with Placement Constraints Assigned

A Pblock with location constraints assigned can be moved. All placement constraints are reassigned within the new location.

Pblocks behave differently when location placement constraints are assigned inside of them. The desired move location should contain adequate resources to assign the placement constraints. The cursor will indicate legal placement sites as the Pblock is being dragged for a move. If there are not adequate resources, a dialog box is displayed, prompting you to either remove or leave the location constraints intact. Fixed and unfixed location constraints are listed separately in the dialog box which enables you to handle them differently.

Locking Placement During ISE Implementation

Placement constraints assigned in PlanAhead are designated as fixed, and when exported they will result in the locked placement during subsequent exported ISE attempts. PlanAhead provides several ways to selectively clear placement constraints that provide control over which LOCs are left in place. The Clear Placement Constraints wizard enables you to clear placement constraints by Pblock, by selection, or all but selected placement constraints. You may also take advantage of Pblock-based implementation and lock the placement for selected Pblocks.

Use the **Fix Instances** popup command to fix logic.

Setting Prohibits

Prohibit constraints can be created for any logic site on the device. To do so:

1. Select the desired sites in the Device view.
You can use Select Area to select more than one site, as described in [“Using the Select Area Command.”](#)
2. Select the **Set Prohibit** popup menu command.

The prohibited sites display a red X ([Figure 10-50](#)).

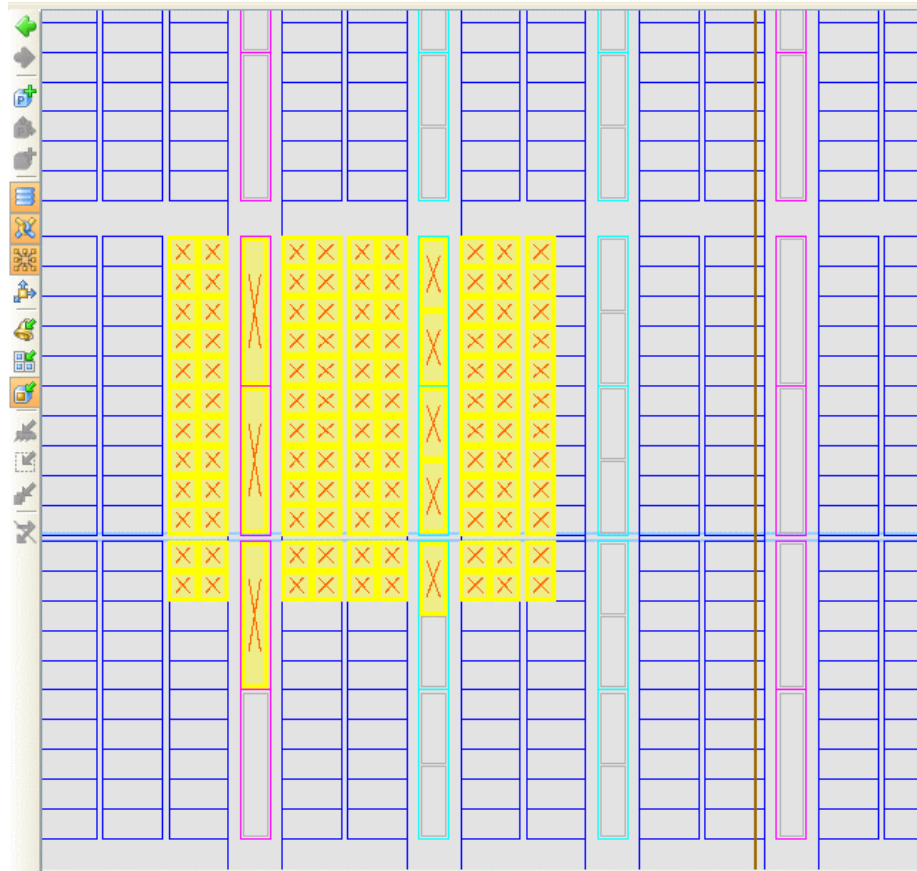


Figure 10-50: Prohibited Sites in the Device View

Using IP Reuse Capabilities

IP Reuse Overview

You can also export and reuse instance modules complete with placement locations and AREA_GROUP constraints. PlanAhead provides the unique capability to export module-level EDIF files and UCFs that contain placement constraints, which can then be imported and reused in other designs. This capability can also be used to facilitate team design or to duplicate placement for identical modules.

Creating and Reusing an IP Module

Exporting an Instance for IP Reuse

In order to facilitate the reuse of placement constraints, they must first be imported into PlanAhead using XDL. For more information, see [“Importing ISE Placement and Timing Results.”](#)

Hierarchical netlist instances are the only items available for exporting as an IP module.

To export the instance-level netlist and placement constraints:

1. Select the instance to be exported from the Netlist view.
2. Select **File > Export IP**.

The Export IP wizard will display.

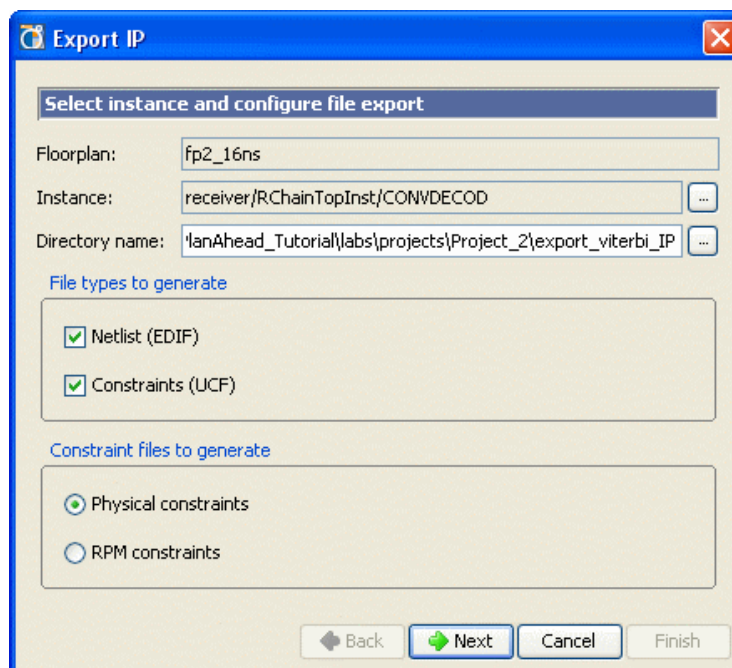


Figure 10-51: Export IP Wizard

3. View and edit the definable fields in the Export IP dialog box:
 - ◆ **Instance**—Enables you to define the instance to export. Use the browse button to select the instance to export. The field can be seeded with a pre-selected instance prior to invoking the “Export IP” command.
 - ◆ **Directory Name**—Displays the directory to export the files, which can be edited. PlanAhead will automatically create a *<instance name>* subdirectory to include the exported files for each instance.
 - ◆ **File types to generate**—Indicates which file types to export: Netlist (EDIF) and/or Constraints (UCF).
 - ◆ **Constraint files to generate**—Indicates which type on constraints to annotate to the output UCF file.
 - **Physical constraints**—Will output physical constraints for all assigned instances.
 - **RPM constraints**—Will replace the LOC constraints with appropriate RLOC and RPM set constraints.
4. Click **Next** to continue.
An Export IP Summary dialog box will display the files that will be created using the command.
5. Click **Finish** to export the files.

Reusing the IP Module

Instance IP files that have been exported can be reused in other designs. The format of the exported IP files maintains the original logic interface of the module which will make reuse much easier. The logic designer should create a black-box module with the exact same interface as the exported instance. If using XST for logic synthesis, the exported EDIF file can be used to derive timing information during synthesis. Once the netlist has been created through synthesis, create a new Project using the new netlist. For more information about creating a new project, see [“Using the Create New Project Wizard to Create a New Project.”](#)

Importing the Instance EDIF Netlist

To reuse an IP module, you must first import the instance EDIF netlist. To import the netlist, use one of the following methods:

- During new project creation—Define the search path in the New Project dialog box to add the module to a new project using **File > New Project**.
- Into existing project—Import the module using the **File > Update Netlist** command after the Floorplan has been created.

For more information on updating a module-level netlist, see [“Updating a Module-Level Netlist.”](#)

Importing the Placement Constraints (UCF)

Once the module netlist has been imported, the next step is to import the placement constraints. To import placement constraints:

1. Select the instance in the Netlist view.
2. Select **File > Import Constraints**.

The selected instance will be seeded in the Import Constraints dialog box.

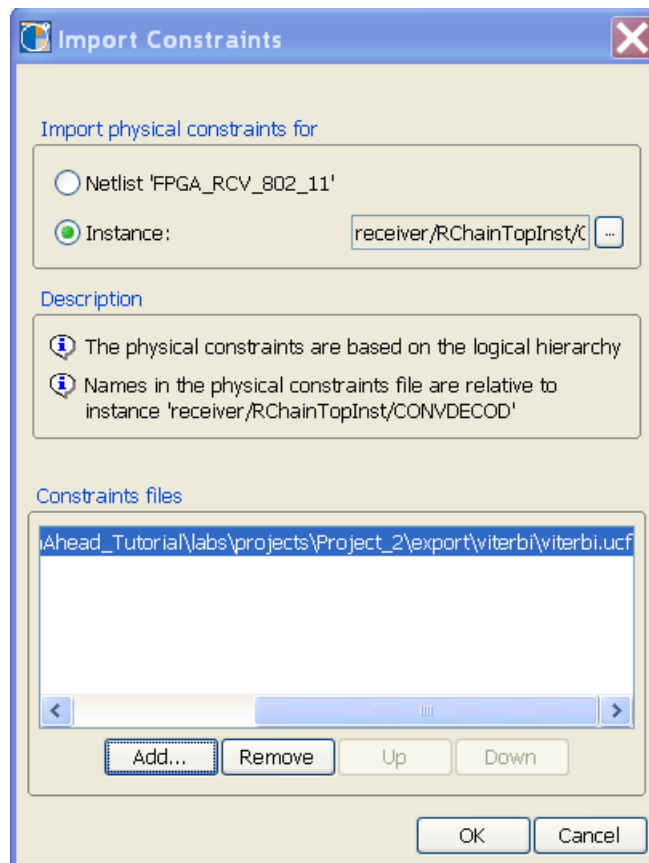


Figure 10-52: Import Module-Level Constraints

3. Set “Import physical constraints for” to **Instance** with the proper instance selected in the field.
4. Use the file browser to select the exported UCF file for the IP instance.
5. Click **OK** to import the constraints.

The physical constraints and AREA_GROUP constraint will be reproduced in the same location as they were created.

Note: The site locations must exist in the new target device.

Duplicating Placement for Identical Modules

Placement constraints can be duplicated for identical modules by selecting the instances individually and using the **File > Import Constraints** command, as described in “[Reusing the IP Module](#).” Once the constraints are imported for one instance, the Pblock can be moved to allow the second instance to be imported in the same location. Repeat the procedure for each module.

Debugging the Design with ChipScope

This chapter contains the following sections:

- “Overview of ChipScope Integration in PlanAhead”
- “Requirements and Limitations When Using Core Insertion Flow”
- “Using the Core Insertion Flow”
- “Implementing the Design”
- “Invoking the ChipScope Analyzer”

Overview of ChipScope Integration in PlanAhead

The ChipScope™ Pro integration provides simplified post-synthesis insertion and connection of the ChipScope Pro ILA debug cores in the PlanAhead™ tool. A GUI wizard is provided for quick and easy design debug for most situations. A non-wizard GUI and Tcl command flow are also available for precision debug core and net connection control. This flow provides a robust ILA core connection solution without leaving the PlanAhead tool.

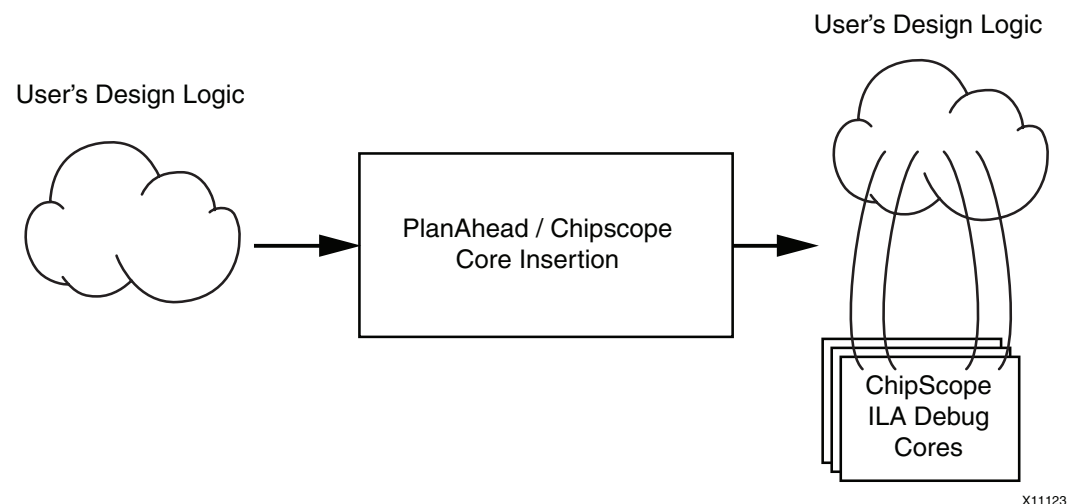


Figure 11-1: Block Diagram of PlanAhead/ChipScope Integration

Requirements and Limitations When Using Core Insertion Flow

PlanAhead/ChipScope integration requires that the Xilinx® ISE® 11.x tools be installed with PlanAhead for ChipScope Pro debug core insertion. The ChipScope Pro 11.1 Analyzer tool and the Xilinx Platform USB cable are required for runtime design debugging. For more information about ChipScope Pro see the *ChipScope Pro 11.1 Software and Cores User Guide* located at:

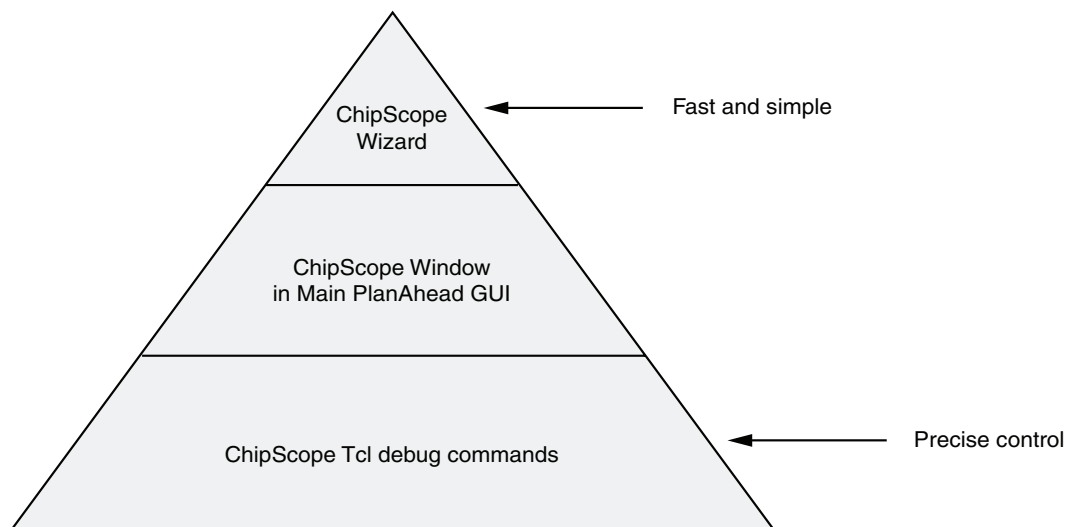
<http://www.xilinx.com/support/documentation/index.htm>

Limitations of the PlanAhead/ChipScope integration are as follows:

- This flow is not compatible with the Project Navigator / ChipScope Pro Core Inserter flow.
- This flow is not available when in ISE Integration mode.
- Pre-existing debug cores connected to a ChipScope Pro ICON core may be viewed but not changed.
- This flow is not compatible with pre-existing ICON generated without a BSCAN primitive that requires connection to a BSCAN primitive instantiated outside of the core.
- Since the PlanAhead tool adds debug cores to the post-synthesis design netlist, some nets may be unavailable for debugging due to trimming or other optimization that takes place during the synthesis process.
- Currently, only the ChipScope Pro ILA cores may be created and connected using this flow.
- Probing inside NGC core files is limited to interface signals only.
- PlanAhead 11, ISE 11, and ChipScope Pro 11 tools must be used with this flow. Mixing and matching tool versions is not supported.

Using the Core Insertion Flow

Insertion of ChipScope debug core in the PlanAhead tool is presented in a layered approach to address different needs of the diverse group of PlanAhead users. The highest level is a simple GUI wizard that automatically creates and configures ILA cores based on the selected set of nets to debug. One level lower is the main ChipScope view allowing control over individual cores, ports and their parameters. The lowest level is the set of Tcl debug commands that may be manually entered or replayed as a script for ultimate control. A combination of the modes can also be used to quickly insert and customize debug cores.



X11122

Figure 11-2: Debug Core Insertion Modes

Deciding Which Debug Core Insertion Mode to Use

Table 11-1 summarizes how to decide what insertion mode(s) to use based on the debugging goal.

Table 11-1: Debugging Goals and Core Insertion Modes

Debugging Goal	Core Insertion Mode
Quickly create ILA debug core(s) with default settings for the selected nets	ChipScope Wizard
Change parameters on existing debug cores	ChipScope Window
Manually create or delete existing debug cores	ChipScope Window
Manually create, delete, or configure trigger or data ports on an ILA core	ChipScope Window
Manually assign nets to the trigger/data/clock channels	ChipScope Window
Play back a recorded script of debug commands later	Tcl Commands

Selecting Nets for Debug

The first step in the PlanAhead/ChipScope debug flow is to identify the set of nets to debug. PlanAhead makes the debug net selection process as simple as picking a set of nets or busses in the Netlist view, and selecting the **Add to ChipScope Unassigned Nets** popup command. Net selection is also available by selecting nets or busses in the Schematic view.

Unassigned Nets List

PlanAhead maintains an Unassigned Nets list (Figure 11-3) in the ChipScope view (**Window > ChipScope**). The Unassigned Nets list is a convenient place to bookmark nets of interest while browsing the design. Nets may be stored for later debug by dragging and dropping from the netlist or schematic view into the Unassigned Nets list. Nets may also

be added to the Unassigned Nets list by selecting the nets, and selecting the **Add to ChipScope Unassigned Nets** popup menu command.

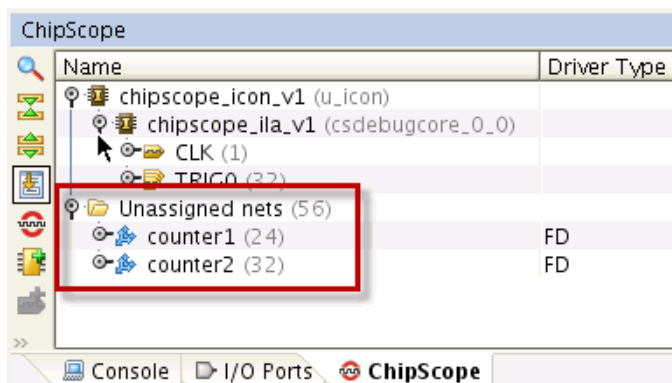


Figure 11-3: Unassigned Nets List in ChipScope Window

ChipScope Wizard-based Debug Core Insertion

The ChipScope debug wizard is the easiest and fastest way to add debug cores in the PlanAhead design tool (Figure 11-4). The first step in using the wizard is to select a set of nets for debug either using the unassigned nets list or direct net selection. The next step is to invoke the wizard using **Tools > Set Up ChipScope**, and to follow the instructions screen by screen to connect and configure the debug cores.

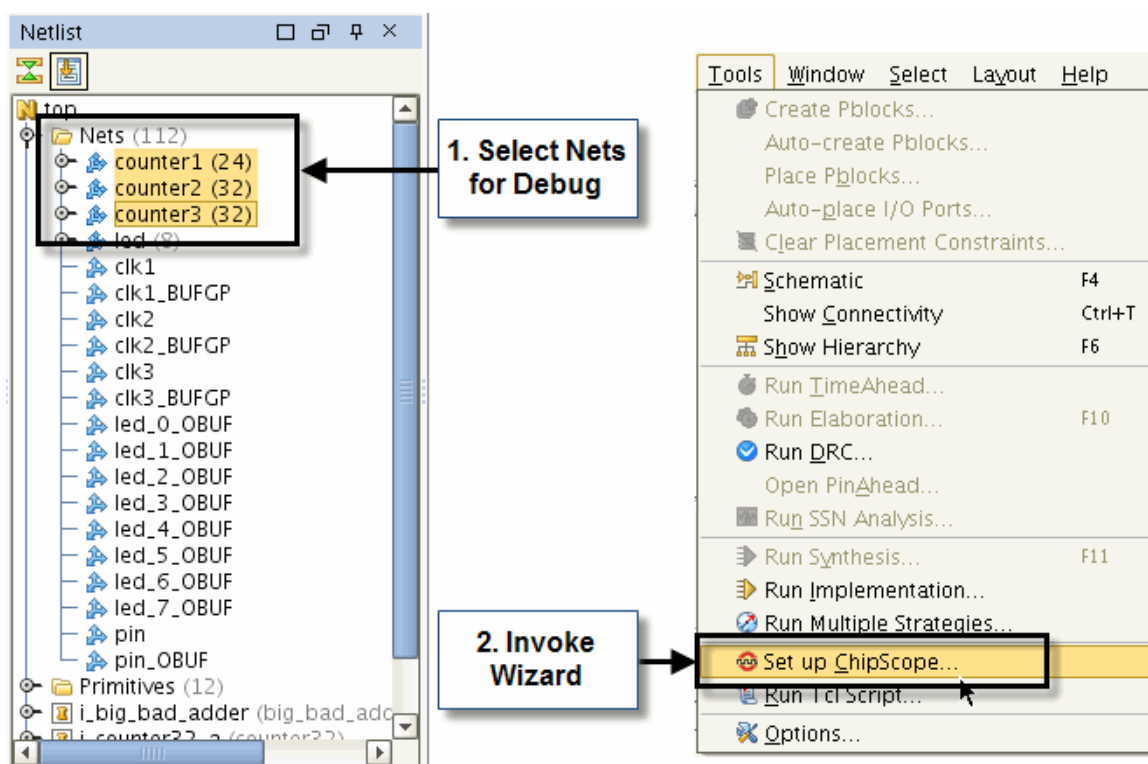


Figure 11-4: Selecting Nets for Debug and Launching the ChipScope Wizard

Specifying Debug Nets and Clock Domains

The ChipScope wizard will attempt to automatically detect the correct clock for each selected net or bus (Figure 11-5). If multiple clocks are detected for a given net, a dropdown list allows the selection of different clocks for the net or bus. The debug net selection may be further modified by clicking the **Add/Remove Nets** button. Each net or bus may be configured for use as a trigger, data storage, or both. When the net and clock configuration is correct, click **Next** to proceed to the summary screen.

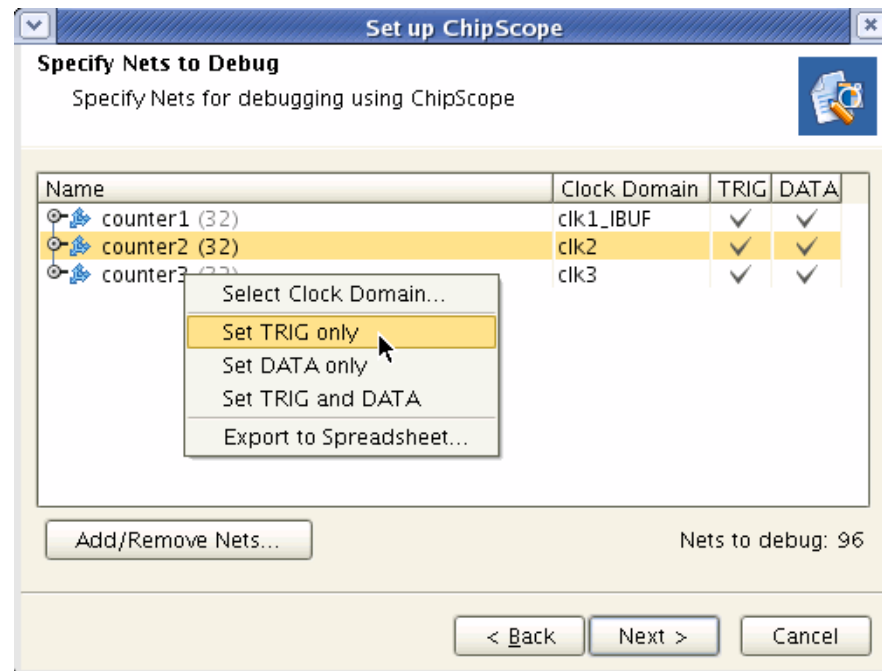


Figure 11-5: Specifying Debug Nets and Clock Domains

Inserting ILA Cores

The ChipScope Wizard inserts one ILA core per clock domain. The nets that were selected for debug are automatically assigned to the trigger and data ports of the instantiated ILA cores. The last wizard screen shows the core creation summary displaying the number of clocks found and ILA cores that will be created and removed (Figure 11-6). If satisfied with the results, click **Finish** to instantiate and connect the ILA cores in the design.

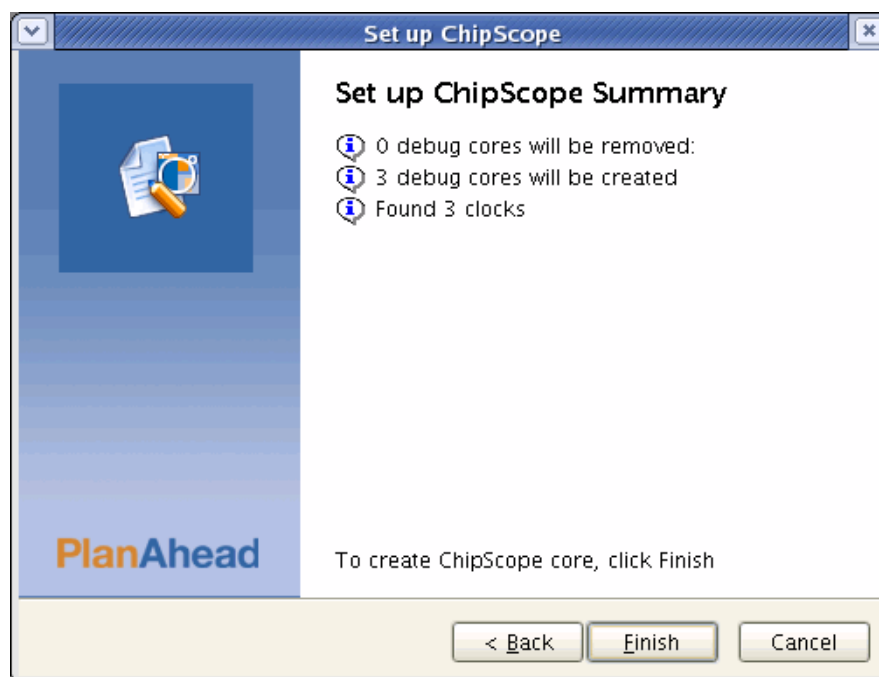


Figure 11-6: Inserting ILA Cores into Design

Using the ChipScope Window to Add and Customize Debug Cores

Fine-grained control over ILA core insertion that is not available in the ChipScope wizard is available in the main ChipScope view. The controls available in this window allow core creation, core deletion, debug net connection, and core parameter changes. To bring up the ChipScope view, select **Window > ChipScope** (Figure 11-7).

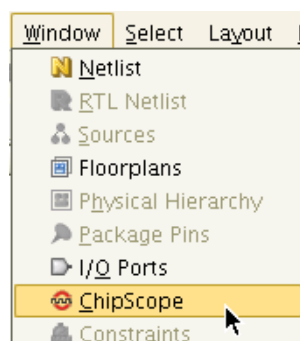


Figure 11-7: Opening the ChipScope Window

The ChipScope Window can also be brought into focus in the foreground by clicking on the ChipScope view tab at the bottom of the PlanAhead main window (Figure 11-8).

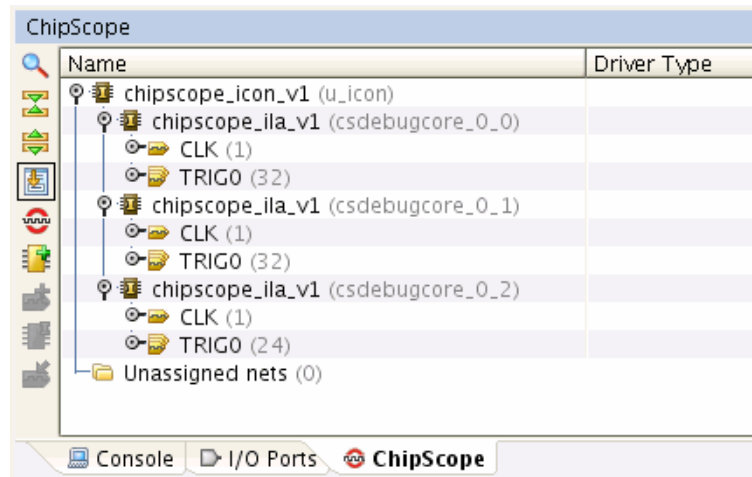


Figure 11-8: ChipScope View

The main ChipScope view shows the list of debug cores connected to the ICON controller core. The ChipScope view also maintains the list of unassigned nets at the bottom of the window. Debug cores and ports may be manipulated from the popup menu or the toolbar buttons on the left side of the view.

Creating and Removing Debug Cores

ChipScope debug cores can be created in the ChipScope view by clicking the **Create ChipScope Debug Core** popup menu command or toolbar button. Using this interface, you can change the parent instance, debug core name, and set parameters for the core (Figure 11-9). To remove an existing debug core, select the core in the ChipScope view, and select the **Delete** popup command.

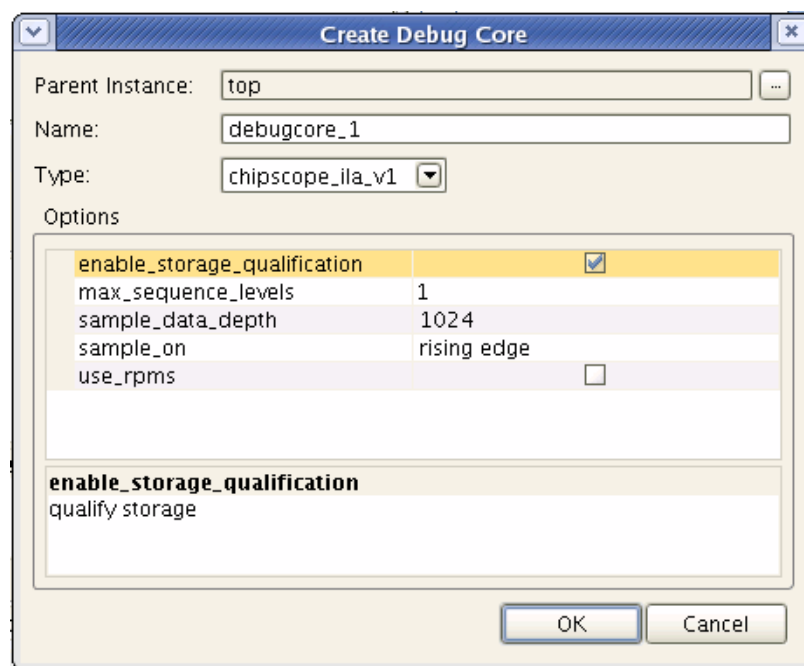


Figure 11-9: Creating a Debug Core

Adding, Removing, and Customizing Debug Core Ports

In addition to adding and removing debug cores, ports of each debug core may be added, removed, and customized to suit your debugging needs. To add a new port:

1. Select the core.
2. Click the **Create ChipScope Debug Port** popup menu command, or toolbar button.

The Create Debug Port dialog box appears.

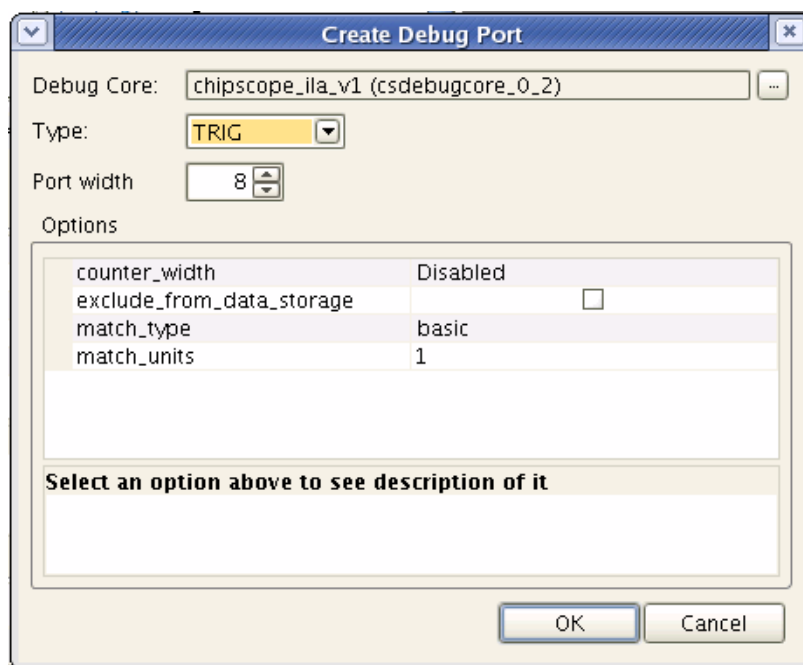


Figure 11-10: Customizing Ports and Options of Debug Cores

3. Select the port type in the dropdown.
Any configurable options for the port will be displayed in the Options area. The port width will start as a default but will expand and contract as nets are added and removed from the port.
4. Click **OK**.

To remove a debug port, in the ChipScope tab select the port, and select **Delete** from the popup menu

Connecting and Disconnecting Nets to Debug Cores

Nets and busses (vectors of nets) may be selected then dragged and dropped from the Schematic view or the Netlist view onto the debug core ports (Figure 11-11). This will automatically expand the port as needed to accommodate the net selection. You can also right click on any net or bus, and select **Assign to ChipScope Debug Port**. To disconnect nets from the debug core port, select the nets that are connected to the debug core port, and select **Disconnect Net** from the popup menu.

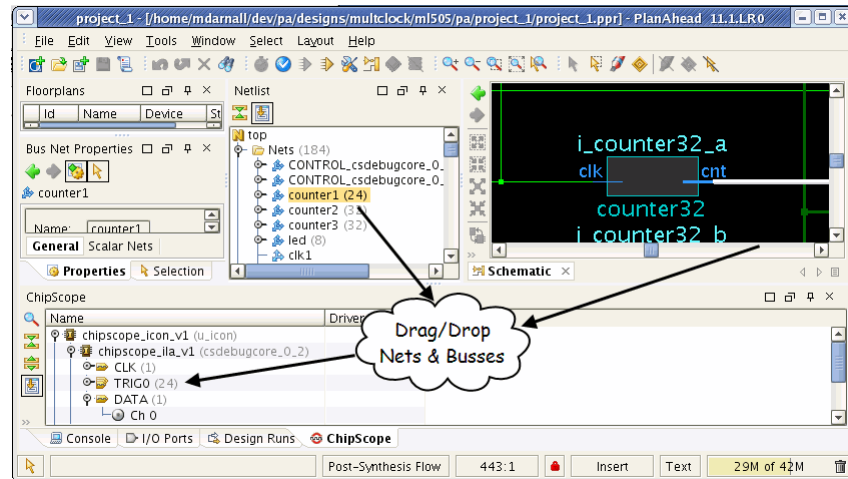


Figure 11-11: Dragging and Dropping Nets onto Debug Core Ports

Customizing Debug Core and Port Parameters

ChipScope Debug cores have parameters that can be customized. To access these core parameters, select one of the ChipScope debug cores in the ChipScope view. In the Properties tab, select **Options** to configure the core parameters (Figure 11-12). Port parameters are modified by first clicking the Trigger or Data port of a debug core, and selecting Options in the Properties tab (Figure 11-12).

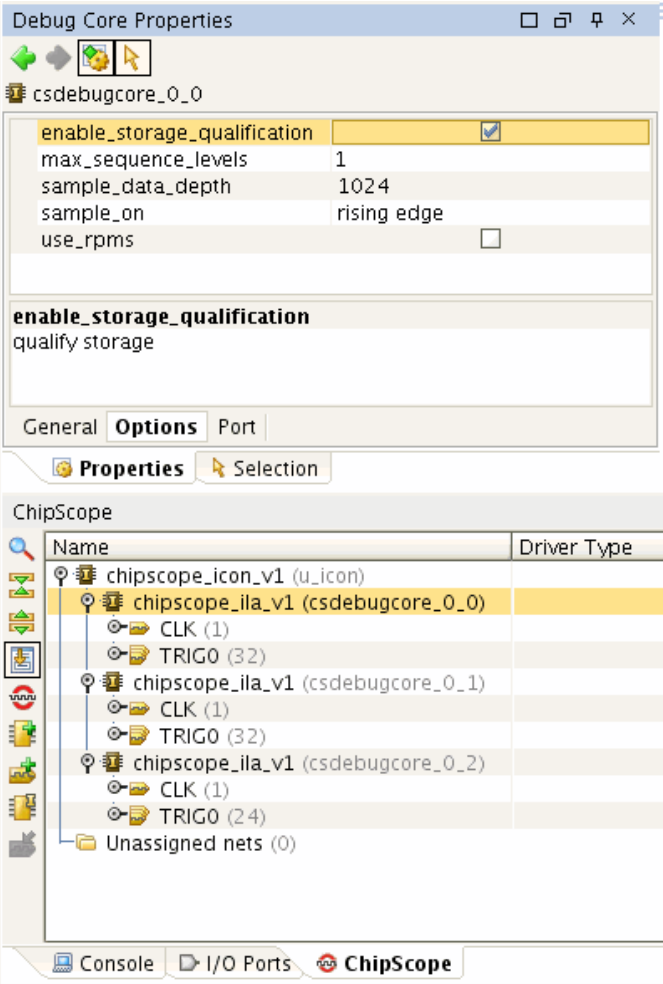


Figure 11-12: Debug Core Parameters

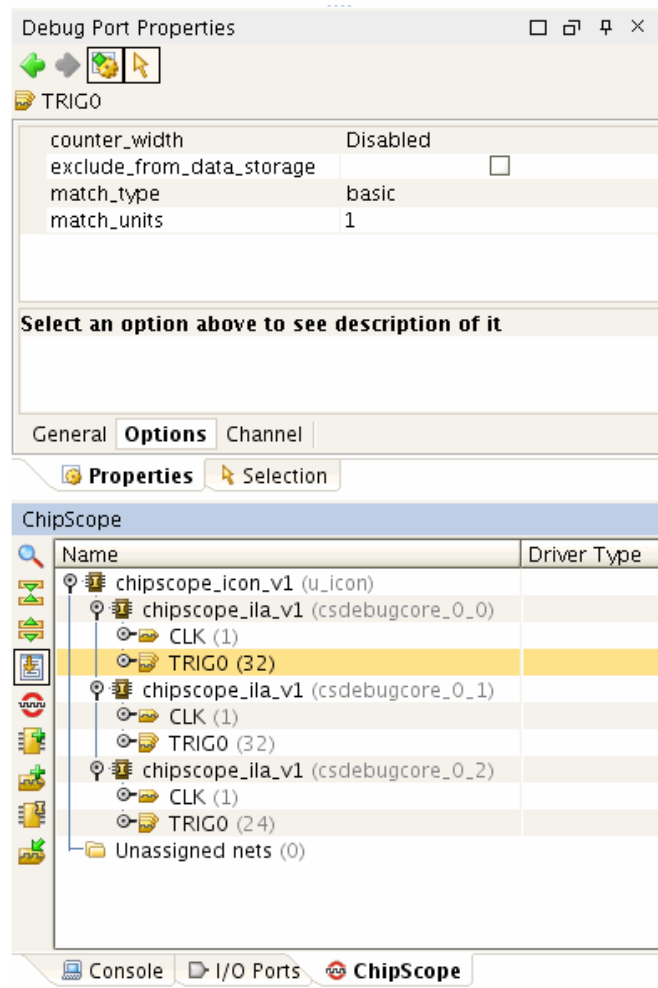


Figure 11-13: Debug Port Parameters

Implementing Debug Cores

ChipScope Pro ICON and ILA cores are initially created in the PlanAhead tool as black boxes. These cores must be implemented prior to running through map, place, and route. ChipScope debug core implementation is automatic when running the implementation flow using **Tools > Run Implementation**. However, you can manually force debug core implementation for floorplanning or timing analysis by clicking the Implement toolbar button on the left side of the ChipScope view. The Xilinx CORE Generator tool will be invoked in batch mode for each black box debug core. This operation may take some time. A progress dialog will indicate the operation is running (Figure 11-14). When debug core implementation is complete, the debug core black boxes are resolved and you may push inside the generated instances.

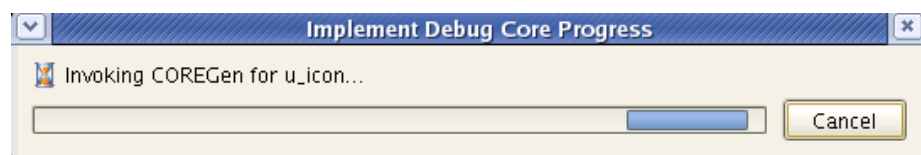


Figure 11-14: Debug Core Implementation Progress Indicator

Exporting Net Connections CDC File for ChipScope Analyzer Tool

A ChipScope Analyzer CDC file is automatically generated when design implementation is complete. You may also manually export a CDC file from the **Export Debug Net Names** popup command in the ChipScope view. This CDC file may be imported into the ChipScope Analyzer to automatically set up the net names on the ILA core data and trigger ports.

Implementing the Design

Once ChipScope debug cores are created and connected, the standard PlanAhead implementation flow may be run to create a bitstream for the device. The implementation flow may be started by selecting **Tools > Run Implementation**.

Invoking the ChipScope Analyzer

If the ChipScope Pro Analyzer software is installed, it may be launched directly from PlanAhead. Before launching the ChipScope Pro Analyzer tool, be sure a current bitstream has been created for the design by right clicking on an implementation run and selecting **Run Bitgen** from the popup menu. To launch ChipScope Pro Analyzer, do one of the following:

- In the ChipScope view tab, right-click and select **Launch ChipScope Analyzer** from the context menu.
- In the Design Runs view tab, right-click and select **Launch ChipScope Analyzer** from the context menu.

The BIT bitstream and CDC netlist name files are automatically passed to the ChipScope Pro Analyzer when launched from PlanAhead.

Menu and Toolbar Commands

This appendix provides a quick reference to the following PlanAhead™ software commands:

- [“Main Menu Commands”](#)
- [“Toolbar Commands”](#)

Main Menu Commands

The following tables provides a quick reference for the PlanAhead software main menu commands. View-specific popup menu commands are covered elsewhere in this document.

Some of the commands are available only during the appropriate PlanAhead mode. Unavailable commands are shown as disabled or “grayed out”.

File Menu

The File menu has the following menu commands:

Table A-1: File menu commands

Command	Description
New Project	Invokes the Create New Project Wizard to facilitate creating of a new Project.
Open Project	Opens an exiting Project file that has previously been saved to disk from PlanAhead. Opening a Project will also open all previously opened Floorplans associated with the Project. It will also restore the latest status of all ISE® software implementation runs.
Reopen Project	Provides a list of previously opened Project to select for reopening.
Save Project	Enables saving of the current Project.
Save Project As	Enables saving of the current Project to another name or location.
Close Project	Closes the active Project. You will be prompted to save any unsaved changes.
New Floorplan	Invokes the New Floorplan Wizard to facilitate creating of a new Floorplan.
Reopen Floorplan	Provides a list of previously opened Floorplans to select for reopening.
Save Floorplan	Saves the Floorplan in the Project. You will be prompted to save any unsaved changes.
Save Floorplan As	Saves Floorplan to a specified name and location. The original Floorplan remains stored as a closed Floorplan in the Project.

Table A-1: File menu commands

Copy Floorplan	Enables copying of the Floorplan and applicable physical constraints to a new target device.
Close Floorplan	Closes the active Floorplan or a group of selected Floorplans in the Project. Closed Floorplans are stored in the Project and are available for reopen in the Floorplans view.
Revert Floorplan	Discards all operations performed in the Floorplan since the last save, and reverts the Floorplan to its original state.
Delete Floorplan	Closes the active Floorplan or a group of selected Floorplans in the Project, and removes the Floorplan data from disk.
Add Sources	Enables the adding of source files or entire directories to the Project.
Create Source	Opens the New Source File dialog box, which enables you to create a source file.
Import I/O Ports	Imports port list from a Comma Separated Values (CSV) or HDL file for pin assignment with PinAhead.
Import Constraints	Imports constraints for the displayed Floorplan. A file selection dialog box appears to select a constraints file (UCF format).
Import Placement	Imports ISE placement results (XDL format) for either an individual Pblock or the entire design.
Import TRCE Results	Imports results from Xilinx® TRCE reports.
Update Netlist	Invokes Update Netlist Wizard for module-level netlist updates.
Export Netlist	Invokes the Export Netlist dialog box to export a EDIF format netlist file to use in ISE.
Export Constraints	Invokes the Export Constraints dialog box to export floorplan constraints to use in ISE.
Export Pblocks	Exports a matching netlist and physical constraint file for a selected Pblock or Pblocks.
Export IP	Exports a logic instance in logical format along with the placement constraints facilitating hard IP reuse. Placement constraints can be created using LOC, BEL, RLOC, and SLICE constraints.
Export I/O Ports	Exports a Comma Separated Values (CSV) format, User Constraints File (UCF) format, or HDL file port list from PinAhead for use in PCB schematic symbol or RTL Header generation.
Print	Prints the current view. This command is only available when the Schematic view, Device view, Package view or Instance Hierarchy view is active.
Exit	Closes the application.

Edit Menu

The Edit menu has the following menu commands:

Table A-2: Edit menu commands

Command	Description
Properties	Invokes the Properties view, which displays more information about the currently selected object.

Table A-2: Edit menu commands

Delete	Removes the currently selected object(s).
Unplace	Unplaces the selected primitive instances and/or I/O ports.
Undo	Reverses the previously run Tcl command in active PlanAhead session.
Redo	Reverses the last Undo command action.
Cut	Cuts the selected text from the HDL Editor into the clipboard.
Copy	Copies the selected object into the clipboard. Currently only floorplans and timing constraints can be copied in PlanAhead.
Paste	Pastes the contents of the clipboard.
Find	Invokes the Find dialog box to search for specific instances or nets.
Find in Files	Invokes the Find in Files dialog box to search for text strings in the selected files.

View Menu

The View menu has the following menu commands:

Table A-3: View menu commands

Command	Description
Zoom In	Increases the scale factor of the active graphical window by a factor of 200%.
Zoom Out	Decreases the scale factor of the active graphical window by a factor of 50%.
Zoom Fit	Fits the entire contents of the active graphical view.
Zoom Area	Invokes Zoom area mode allowing the drawing of a zoom rectangle in the active view.
Fit Selection	Adjust the zoom level to fit all of the selected items in the active window.
Fit Highlight	Adjust the zoom level to fit all of the highlighted items in the active window.
Fit Markers	Adjust the zoom level to fit all of the marked items in the active window.
Refresh	Redraws the graphics in the active window.

Tools Menu

The Tools menu has the following menu commands:

Table A-4: Tools menu commands

Command	Description
Create Pblocks	Invokes the Create Pblocks wizard seeded with selected set of instances to allow creation of multiple Pblocks.
Auto-create Pblocks	Invokes the Auto-create Pblock dialog box to automatically place the instances into Pblocks and name them accordingly.
Place Pblocks	Invokes the Place Pblocks dialog box to automatically size and place Pblocks.
Auto-place I/O Ports	Places the entire device or any selected portion of it, while obeying I/O bank rules, differential pair rules, and global clock pin rules. A PinAhead feature.

Table A-4: Tools menu commands

Clear Placement Constraints	Invokes the Clear Placement Constraints Wizard to selectively remove port or placement location constraints.
Schematic	Opens a new Schematic view in the Workspace and displays a schematic of the currently selected elements.
Show Connectivity	Selects the elements connected to the selected instances, nets, or Pblocks. This command can be used sequentially to continue to fanout and select a cone of logic.
Show Hierarchy	Invokes a Hierarchy view in the Workspace and graphically displays the entire logic hierarchy. Selected logic is highlighted in the tree.
Run TimeAhead	Invokes the Run TimeAhead dialog box and enables you to configure and launch the TimeAhead static timing analyzer.
Run Elaboration	Invokes the RTL parsing and elaboration, which enable resource estimation and RTL schematic exploration.
Run DRC	Invokes the Run DRC dialog box and enables you to configure and launch the PlanAhead design rule checker
Open PinAhead	Opens the PinAhead environment layout, which enables you to define I/O pinout configurations to meet design and device I/O requirements.
Run WASSO Analysis	Invokes the Run WASSO Analysis dialog box, and generates a WASSO analysis report.
Run SSN Analysis	Invokes the Run SSN Analysis dialog box and generates a Simultaneous Switching Noise (SSN) report of potential I/O issues. Virtex®-6 designs only.
Run Synthesis	Invokes the Run Synthesis dialog box to create and launch a Synthesis Run.
Run Implementation	Invokes the Run Implementation dialog box to run a implementation strategy using one or more hosts (Linux only).
Run Multiple Strategies	Invokes the Run Multiple Strategies dialog box to set up for synthesis runs.
Run Tcl Script	Invokes the Run Script dialog box to execute a Tcl script selectable through file browser.
Options	Invokes the Options dialog box for setting display options, selection options, shortcut options, strategies, etc.

Window Menu

The Window menu has the following menu commands:

Table A-5: Window menu commands

Command	Description
Netlist	Displays the Netlist view.
RTL Netlist	Displays the RTL view.
Sources	Displays the Sources view.
Floorplans	Displays the Project Floorplans view.
Physical Hierarchy	Displays the Physical Hierarchy view.

Table A-5: Window menu commands

Package Pins	Displays Package Pins view.
I/O Ports	Displays I/O Ports view.
ChipScope	Displays the integration ChipScope view.
Constraints	Displays the Constraints view.
Clock Regions	Displays the Clock Regions view.
Metrics	Displays the Metric view.
Selection	Displays the Selection view.
World	Displays the World view.
Console	Displays the Console view.
Timing Results	Displays the timing results in the Timing Results view.
DRC Results	Displays the DRC results in the DRC Violations browser.
WASSO Results	Displays the WASSO Results view.
SSN Results	Displays the SSN Results view.
Find Results	Displays the found objects in the Find Results view.
Find in Files Results	Displays the found objects in the Find in Files view.
Elaboration Log	Displays the results log of the Elaborate Design command.
Design Runs	Displays the runs in the Design Runs view.
Metric Results	Displays the Metrics Results view.
New Device View	Opens a new Device view in the Workspace.
New Package View	Opens a new Package view in the Workspace.
View Log File	Opens the <code>planAhead.log</code> log file, which captures the contents of the messages created when running PlanAhead commands.
View Journal File	Opens the <code>planAhead.jou</code> journal file, which cumulatively captures all of the TCL commands from PlanAhead sessions that were invoked.

Select Menu

The Select menu has the following menu items:

Table A-6: Select menu commands

Command	Description
Unselect All	Unselects all selected elements.
Unselect Type	Unselects elements of a particular type.
Select Area	Invokes the Select Area command putting the cursor in draw rectangle mode in the active Workspace view.
Highlight	Highlights all selected objects using the active highlight color.
Unhighlight All	Unhighlights all highlighted objects.

Table A-6: Select menu commands

Unhighlight	Unhighlights selected objects.
Unhighlight Color	Unhighlights elements based on object color.
Mark	Marks selected objects in the Device view.
Unmark	Unmarks current selected object in the Device view.
Unmark All	Unmarks all marked objects.
Show Source	Invokes the HDL Editor highlighting the source of the selected logic
Show Definition	Restricted to RTL instances of non-primitives, invokes the HDL Editor highlighting the definition of the Verilog module/VHDL entity that is being instantiated.

Layout Menu

The Look and Feel menu has the following menu items:

Table A-7: Layout menu commands

Command	Description
Load Default Layout	Loads the user-defined default view layout, if available.
Save as Default Layout	<p>Saves the current view configuration as the default to be used each time PlanAhead is invoked. This layout will be used each time PlanAhead is invoked.</p> <p>On Windows, the file is saved to the following area:</p> <pre>C:\Documents and Settings\<username>\Application Data\HDI\layouts\application_layout\default.layout</username></pre> <p>On Linux or Solaris, the file is stored in the following area:</p> <pre>~/.HDI/layouts/application_layout/default.layout</pre>
Clear Default Layout	Clears the current user-defined default layout.
Save Layout As	<p>Enables you to save the current view configuration with a user defined name. These layouts can be loaded manually each time PlanAhead is invoked. The area is located on Windows in the folder shown below.</p> <p>On Windows, the layout files are saved to the following area:</p> <pre>C:\Documents and Settings\<username>\Application Data\HDI\layouts\floorplan_layout\<layoutname>.layout</layoutname></username></pre> <p>On Linux or Solaris, the files are stored in the following area:</p> <pre>~/.HDI/layouts/application_layout/<layoutname>.layout</pre>
Load Layout	Enables you to load any of the previously saved layouts, the default PlanAhead layout, or the supplied PlanAhead alternate layouts.
Remove Layout	Enables you to delete any of the previously saved user-defined layouts.
Undo	Reverses the last view manipulation command.
Redo	Reserves the last Undo command action.

Help Menu

The Help menu has the following menu items:

Table A-8: Help menu commands

Name	Description
PlanAhead User Guide	Invokes the PlanAhead User Guide in a separate window.
PlanAhead Methodology Guide	Invokes the PlanAhead Methodology Guide in a separate window.
Tutorial	Enables you to invoke the available PlanAhead tutorials in a separate window.
Release Notes	Invokes the current Release Notes in a separate window.
Check for Updates	Checks the Xilinx website for software updates and, if found, prompts you to install the updates.
License	Invokes the Xilinx License Configuration Manager (XLCM) to locate or manage the software licenses. FLEXnet Publisher is now used for software licensing. XLCM can identify a license or help with the license flow.
PlanAhead on the Web	Opens the Xilinx PlanAhead website in your default browser.
Getting Started	Invokes the Getting Started jump page.
About PlanAhead	Displays information about PlanAhead version and copyright information.

Toolbar Commands

PlanAhead has a fixed toolbar that contains the most commonly used commands. The toolbar button and shortcut key (if available) are displayed for each command.

Table A-9: Toolbar commands









Toolbar	Command Name	Shortcut Key	Description
	New Project		Opens the “New Project” wizard.
	Open Project		Opens the “Open Project” dialog box.
	New Floorplan		Opens the “New Floorplan” wizard.
	Save Project	Ctrl+S	Saves the Project with the current Project name.
	Run Tcl Script		Invokes the Run Script dialog box to select and execute a Tcl script.
	Undo	Ctrl+Z	Undoes previous run command from active PlanAhead session.
	Redo	Shift Ctrl+Z	Redoes previously undone command from active PlanAhead session.
	Delete	Delete	Deletes the current selection.

Table A-9: Toolbar commands






















	Find	Ctrl+F	Invokes the Find dialog box to search for specific design elements.
	Run TimeAhead		Invokes the TimeAhead dialog box to run static timing analysis.
	Run DRC		Invokes the DRC dialog box to run the design rule checker.
	Run Synthesis	F11	Invokes the Run Synthesis dialog box to create and launch a Synthesis Run.
	Run Implementation		Invokes the Run Implementation dialog box to run a implementation strategy using one or more hosts (Linux only).
	Options		Invokes the PlanAhead Options dialog box.
	Schematic	F4	Invokes a new Schematic view in the Workspace area displaying the pre-selected elements.
	New Device View		Opens a new Device view in the Workspace area.
	Clear Placement Constraints		Open the Clear Placement Constraints dialog box. This dialog box is seeded based on pre-selected objects.
	Zoom In	Ctrl+I	Enlarges the display in the active window.
	Zoom Out	Ctrl+O	Shows more of the display in the active window.
	Zoom Area	Ctrl+R	Zooms into a user defined rectangular area.
	Zoom Fit	Ctrl+G	Fits the active window view.
	Fit Selection	F9	Fits the active Workspace view to display all selected objects
	Select Mode		Enables normal select mode.
	Select Area		Enables menu select mode.
	Highlight		Highlights a selected object.
	Mark	Ctrl+M	Marks selected objects in the Device view.
	Unhighlight All	Ctrl+K	Unhighlights all highlighted objects.

Table A-9: Toolbar commands

	Unmark All		Removes markers from all marked objects.
	Unselect All	F12	Unselects all selected objects.

Installing Releases with XilinxUpdate

Note: This appendix applies to the PlanAhead™ full version only.

This appendix contains the following sections:

- [“PlanAhead Release Strategy”](#)
- [“Running XilinxUpdate”](#)
- [“Automatically Checking for Updates”](#)

PlanAhead Release Strategy

The PlanAhead software release strategy is the same as other Xilinx® software tools. PlanAhead is a tool that is targeted to introduce new technology and respond to customer. New releases are periodically introduced. The version number reflects the release (e.g. 11.1, 11.2, etc.). The **Help > About PlanAhead** command will display the currently installed PlanAhead version.

To check for new PlanAhead releases, run the **Help > Check for Updates** command, as described below.

Refer to the *Xilinx ISE Design Suite: Installation, Licensing, and Release Notes* for more information about Xilinx tool installation.

Running XilinxUpdate

PlanAhead now invokes the *XilinxUpdate* utility to download and install new releases. The utility automatically searches all tools that are installed, compares them with the newest updates on the Xilinx website, and notifies the user of any available updates. Users can selectively install updates for any Xilinx tool.

To check for and install PlanAhead updates, select the **Help > Check for Updates** command.

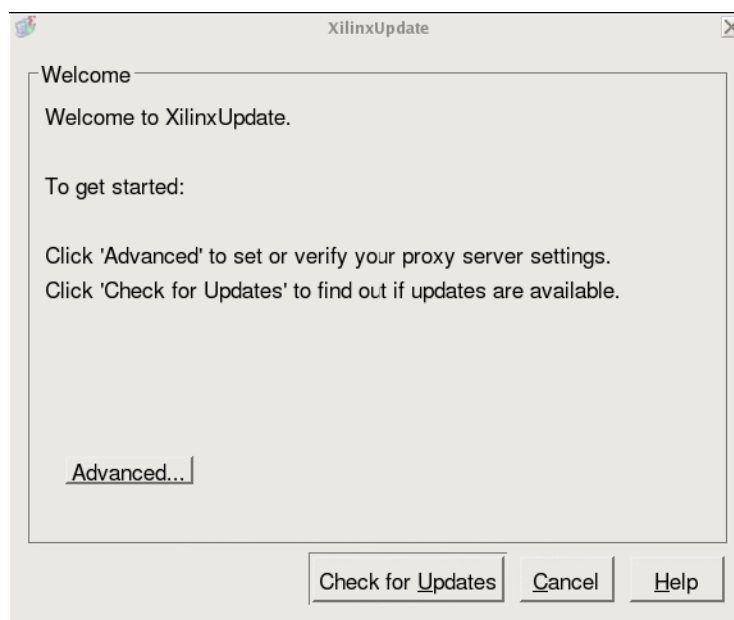


Figure B-1: XilinxUpdate Welcome page

The **Advanced** button will allow specific proxy settings to be validated or set.

1. Select the **Check for Updates** button to communicate with the Xilinx website and return a list of all product updates available.
2. In the Available Updates dialog box, select the desired tool updates.
3. Click **OK** to download and install any updates.

Automatically Checking for Updates

PlanAhead can be configured to automatically search for new incremental releases each time PlanAhead is invoked. To do so, set the **Automatically check xilinx.com for software updates on startup** option in the Tools > Options> General dialog box, as shown below.

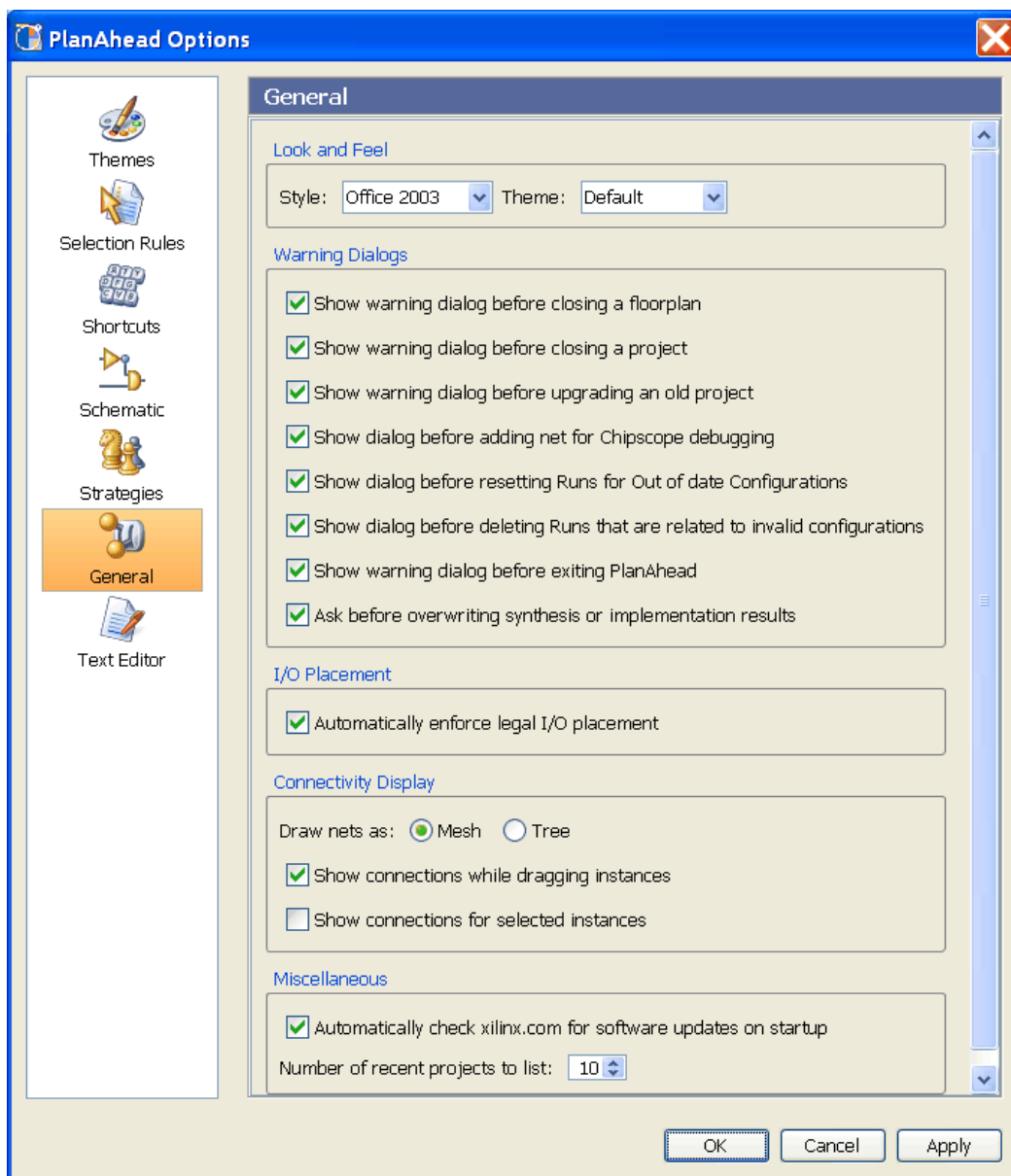


Figure B-2: Automatically Check xilinx.com for Software Updates

Configuring SSH Without Password Prompting

This appendix contains the following section:

- [“Setting Up SSH Key Agent Forward”](#)

Setting Up SSH Key Agent Forward

The multiple host capabilities for executing PlanAhead™ software runs uses Secure Shell (SSH), a service provided by Linux operating system, and not PlanAhead. Prior to configuring multiple hosts in PlanAhead, you must configure SSH so that you are not prompted for a password each time you log in to a remote machine. SSH configuration is accomplished with the following commands at a Linux terminal or shell.

Note: This is a one-time step, and once successfully set up does not need to be repeated.

1. Run the following command at a Linux terminal or shell to generate a public key on your primary machine. Though not required, you should enter (and remember) a private key phrase when prompted for maximum security.

```
ssh-keygen -t rsa
```

2. Append the contents of your publish key to an `authorized_keys` file on the remote machine. The `remote_server` below should be changed to a valid host name:

```
cat ~/.ssh/id_rsa.pub | ssh remote_server "cat - >> ~/.ssh/authorized_keys"
```

3. Run the following command to prompt for your private key pass phrase, and enable key forwarding:

```
ssh-add
```

You should now be able to **ssh** to any machine without typing a password. The first time you attempt to access a new machine, it will prompt you for a password, then subsequent accesses will not. If you are always prompted for a password, contact your System Administrator to have your Linux account set up for passwordless SSH.

After a passwordless SSH is set up, you can continue with [“Configuring Remote Hosts \(Linux Only\).”](#)

Index

A

About

- PlanAhead 5
- PlanAhead-ChipScope integration 329
- PlanAhead-Project Navigator integration 6, 79
- this Guide 5

Add Sources 168

Appendix

- A) Menu and Toolbar Commands 341
- B) Installing Releases with XilinxUpdate 351
- C) Configuring SSH Without Password Prompting 355

B

Block-based design

- Creating and Reusing an IP Module 325
- Exporting Pblocks for implementation in ISE 215
- Implementation Pblocks 194

Bundle Nets. *See* Connectivity

C

ChipScope

- Implementing 340
- Integration 329
- Opening ChipScope Analyzer 340
- Using Core Insertion 330

Clock Regions 225

Close Floorplan 75

Close Project 61

Commands 341

- Common popup menu and view specific toolbar commands 251
- Main menu 341
- Toolbar 347

Connectivity 276

- Adjusting Bundle net defaults 311
- Analyzing hierarchical 258
- Displaying Bundle net properties and content 310
- Expanding logic in the schematic 276, 277
- Using connectivity display to place Pblocks 309

Console view 105

- Using Tcl help 106
- Using the Command line 105

Constraints 68

- DCI_CASCADE 149
- Importing constraints 68
- Importing module-level constraints 69

Constraints view 243

- Adding new timing constraints 246
- Modifying timing constraint values 245
- Removing timing constraints 247

Create a New Project 39

Create I/O Port Interface 141

Create I/O Ports 138

Create Source 170

CSV port list. *See* PinAhead - CSV file

Cursor 111

- Using the Context Sensitive Cursor 111

Custom layout 122

D

DCI_CASCADE constraint 149

Default layout 123

Delete Floorplan 76

Design flow 19

- Basic 20
- Design analysis, floorplanning 21
- Diagram 20
- Run experimentation 20

Design resource utilization statistics 253

- Exporting 254, 257

Design Runs view

- Import Run 205, 207
- Run BitGen 208

Device compatibility 134

Device view 132, 221

- Displaying clock regions 225
- Opening multiple Device views 226
- Placing I/O Ports 143
- Printing 226
- Viewing device resources 223

Display options 112

Documentation

- Understanding terminology 32

DRC 259

I/O Pin and Clock Rules 152, 154

I/O related 128, 143

RTL related 178

Rule Descriptions 154, 180, 262

Running DRC 259

Viewing violations 153, 179, 261

E

Elaborate the RTL design 173

- Using filters 175
- Viewing results 174

error file for debugging

- Xilinx Technical Support 25

F

Find 270

- Searching for objects using the Find command 270
- Using Find in Files 172
- Using the Find Results view 173, 272

Fixed constraints 315

Floorplan Editor

- transition to PlanAhead 86

Floorplan. *See* Project - Floorplan

Floorplanner

- transition to PlanAhead 90

Floorplanning 287

- Connectivity. *See* Connectivity

Manipulating Pblocks 296

- Assigning logic to Pblocks 298
- Creating child Pblocks 293
- Deleting a Pblock 302
- Moving Pblocks 299
- Removing a Pblock rectangle 301
- Renaming a Pblock 302
- Stretching a Pblock 300
- Unassigning logic to Pblocks 299

- Viewing Pblock Properties 302

Partitioning design with Pblocks 288

- Draw Pblock command 288
- New Pblock command 290

- Using New Pblocks command for selected modules 290

Understanding Pblock graphics 296

- Using connectivity display to place Pblocks 309

- Using resource utilization statistics to size Pblocks 307
- Using the automatic Pblock commands 311
 - Running the Partitioner 311
 - Running the Pblock Placer 313

G

- Getting Started 36
- Glossary of terms 32

H

- HDL Editor 171
- HDL format header. *See* PinAhead
- Hierarchy view 236
- Highlighting objects
 - Highlighting Primitives 280
 - Unhighlighting objects 280

I

- I/O nets. *See* Connectivity
- I/O pin planning. *See* PinAhead
- I/O Ports view 129
- Implementation 185
 - ChipScope flow 340
 - Interfacing with ISE 214
 - Importing ISE placement results 216
 - Importing ISE TRCE results 217
 - Pblocks 194
 - Runs 191, 195
 - Importing ISE results for Run 207
 - On Multiple Linux Hosts 211, 355
 - Running Bitgen on Run 208
 - Strategies 185, 209
 - Creating common group strategies 211
 - Creating Strategies 209
 - Modifying ISE command options 211
- Importing
 - Constraints 328
 - I/O Ports 136
- Input files 21
- Instance Hierarchy view 236
- IP Reuse 325
- ISE
 - Implementation 214
 - Placement and Timing Results 273

ISE integration. *See* Project Navigator integration

L

- Launching PlanAhead 35
- Layout
 - Default layout 123
 - Load layout 123
 - PinAhead Cockpit 83
 - PlanAhead Default 85
 - Saving custom layouts 122

M

- Make Part Compatible 134
- Marking objects 110, 281
 - Removing marks 282
- Metrics view 282
 - Configuring Metric ranges 285
 - Displaying Metrics in the Device view 283
 - Hiding Metrics map display 284
 - Using the Metrics Results table 284

N

- Netlist update 61
 - Updating a module-level netlist 62
 - Updating the top-level netlist 61
- Netlist view 239
 - Collapsing the netlist tree 240
 - Selecting logic in the Netlist view 243
 - Understanding the Netlist view icons 242
 - Using the Nets folder 241
 - Using the Primitives folder 241
 - View specific popup menu commands 243
- Nets Display. *See* Connectivity
- New Device View 226
- New Project 39

O

- Open PinAhead 83, 96, 127
- Open Project 60
- Options 112, 233
- Output files 24

P

- PACE
 - transition to PlanAhead 86

- Package Pin Properties 132
- Package Pins view 130
 - Prohibiting I/O Pins
- Package view 131
 - Placing I/O Ports 143
 - Printing 132
 - Prohibiting I/O Pins
 - Show Differential I/O Pairs 132
 - Viewing Package Pin properties 132
- Pblock. *See* Floorplanning
- Physical Hierarchy view 247
 - Understanding the Physical Hierarchy icons 248
 - Using the ROOT Pblock 248
 - Working with Relatively Placed Macros RPMs 249
- PinAhead 125
 - Autoplacing I/O Ports 128, 146
 - Configuring I/O Ports 139
 - Creating I/O Port Interfaces 128, 130, 141
 - Creating I/O Ports 138
 - CSV file
 - Contents 136
 - Exporting 164
 - Importing 136
 - User Columns 137
 - Default Layout 96
 - Device resources 132
 - Device view 132
 - DRC *See* DRC
 - HDL file
 - Importing I/O ports 137
 - I/O Ports view 129
 - Importing I/O Ports 136
 - Layout 84
 - Opening 96, 127
 - Package Pins view 130
 - Package view 131
 - Placing GT I/O Ports 148
 - Placing I/O Ports 142
 - Enabling DRC 143
 - Into I/O Banks 143
 - Sequentially 145
 - To Defined Area 144
 - Placing I/O related clock logic 148
 - Prohibiting I/O Pins 140
 - Running SSN analysis 157
 - Running WASSO analysis 161
 - UCF file
 - Importing I/O ports 138
 - Viewing Clock Region resources 134

- Viewing I/O bank resources 133
- Place and route. *See* Implementation
- Placement constraints
 - Adjusting the visibility of LOC constraints 317
 - Assigning BEL constraints 316
 - Assigning LOC or Site constraints 316
 - Deleting selected LOC/BEL constraints 319
 - Duplicating placement for identical modules 328
 - Importing ISE placement results 216
 - Locking placement during ISE implementation 324
 - Moving 318
 - Pblocks with placement constraints assigned 324
 - Selectively clearing LOC constraints 319
 - Setting Prohibits 324
 - Understanding fixed and unfixed constraints 315
 - Understanding LOC and BEL constraints 315
 - Viewing primitive logic connectivity 318
- PlanAhead configuration
 - Adjusting Bundle net defaults 311
 - Checking for software updates 351
 - Configuring tool shortcut keys 118
 - Setting display options 112
- PlanAhead default layout 85
- PlanAhead inputs 21
 - Constraint files 22, 23
 - CSV files 22
 - HDL files 23
 - ISE placement results 23
 - ISE TRCE timing results 24
 - Module-level netlists and cores 23
 - RTL files 22
 - Top-level netlists 23
 - Xilinx cores 22
- PlanAhead Options 112
- PlanAhead outputs 24
 - Command logs and reports 24
 - Command journal file 25
 - Command log file 24
 - Design logical netlist and Pblock statistics report files 25
 - DRC results files 25
 - Error report file 25
 - TimeAhead results and constraints files 25

- WASSO analysis reports 26
- CVS files 24
- Environment defaults 26
 - Initialization file 26
 - ISE implementation strategy files 27
 - Shortcuts definitions 27
 - View layout defaults 26
- ISE implementation data 28
 - ChipScope core netlists 30
 - Constraint files 31
 - EDIF netlists 29
 - ISE run scripts 31
 - Run directories 29
- Project data 27
 - Floorplan data directories 28
 - Project data directory 28
 - Project directory 27
 - Project file 28
 - Project netlist 28
- RTL files 24
- PlanAhead product description 5
- Printing 103
 - Device view 226
 - Package view 132
 - Schematic view 232
- Prohibit Pins. *See* PinAhead
- Project 38
 - Adding sources 168
 - Closing a Project 61
 - Copying sources into directory 169
 - Creating new source 170
 - Floorplan 65
 - Closing a floorplan 75
 - Copying a floorplan 77
 - Creating a floorplan 65
 - Deleting a floorplan 76
 - Renaming a floorplan 77
 - Saving a floorplan 76
 - Viewing and editing floorplan properties 73
 - Importing constraints 68
 - Importing module-level constraints 69
 - Importing top-level constraints 68
 - Netlist update 61
 - Updating a module-level netlist 62
 - Updating the top-level netlist 61
 - Opening a Project 59
 - Opening an existing project 59

- Opening multiple projects 60
 - Using the Getting Started jump page 38
- Starting a Project 35
 - Using the Create Project Wizard 39
 - Using the Getting Started jump page 38
- Types 38
 - Empty 38, 40
 - Implementation results-based 39, 40
 - RTL-based 38, 40
 - Synthesis netlist-based 38, 40
- Project Navigator integration 79
 - Analyze Timing/Floorplan Design (Post-Implementation) 82, 86
 - Floorplan Area/IO/Logic (Post-Synthesis) 82
 - I/O Pin Planning (Post-Synthesis) 81
 - I/O Pin Planning (Pre-Synthesis) 81
- Properties view 238

R

- Recommended Pin Planning Method 125
- Remote Hosts (Linux)
 - Configuring 211, 355
- Resource utilization
 - Estimates for modules 176
 - Statistics 253
 - Using utilization statistics to size Pblocks 307
- resource utilization statistics
 - Exporting 254, 257
- RTL Editor 171
 - View-specific popup menu commands 172
- RTL Hierarchy view 176
- RTL Netlist view 175
 - resource estimates 176
 - View-specific popup menu commands 176
- RTL Schematic view 177
- Run Elaboration 173
- Run Implementation 191, 195
- Run Multiple Strategies 188, 193
- Run Synthesis 185
- Run Tcl Script 37
- Running DRC 178
- Running PlanAhead 35

S

Save Floorplan 76

Schematic view 227

- Annotating fanout and slack on schematic pins 233
- Expanding and collapsing logic for selected modules 230
- Expanding logic from selected pins 229
- Printing 232
- Regenerating schematic 231
- Selecting objects 232
- Traversing hierarchy 231
- Viewing logic hierarchy 228
- Viewing timing path logic 236
- View-specific popup menu commands 232

Selecting objects 108

- Fitting display to show selected objects 109
- Highlighting objects. *See* Highlighting objects
- Marking objects. *See* Marking objects
- Selecting multiple objects 108
- Selecting objects using the Find command 270
- Selecting Primitive Parents 109
- Selecting Primitives 280
- Setting selection ability for objects 111
- Setting selection rules 110
- Using Select Area 108
- Using the Select menu commands 109

Selection view 109

Software update 351

- Automatic check for updates 120

Source File Properties 168

Sources view 165

- View-specific popup menu commands 167

SSN analysis 157

- Running SSN analysis 157
- Viewing results 158

Starting PlanAhead 35

- Command line options 36
- Invocation 35
- Using a startup Tcl script 37
- Using the Getting Started jump page 38

Synthesis 185

- Design resource utilization statistics 253
- Exporting 254, 257

Runs 185

- Importing results for Run 205
- On Multiple Linux Hosts 211, 355

Strategies 185, 209

- Creating common group strategies 211
- Creating Strategies 209
- Modifying command options 211

Synthesis Options 174

T

Timing Analysis 264

- Excluding paths from analysis 266
- Running TimeAhead 264

Timing Results 266, 274

- Displaying path details 269
- Removing paths from report 268
- Sorting timing report 267
- Viewing path reports in Workspace 270
- Viewing timing paths in Device view 274
- Viewing timing paths in Schematic view 275
- Viewing TRCE timing results 274

U

Undo Dragging 122

Unfixed constraints 315

Unhighlighting objects 280

Update Netlist 61, 62

V

Viewing environment 95

- Common popup menu and view specific toolbar commands 251
- Configuring tool shortcut keys 118

Configuring views

- Auto-hiding views 100
- Defining view locations within viewing areas 121
- Defining viewing area sizes 101
- Manipulating views using view banner commands 100
- Moving views to other viewing areas 121
- Opening views 99
- Resetting views back to the default layout 123
- Restoring views to original locations 123

Saving view layouts 122

Undoing view manipulations 123

Default view layout 95

Selecting objects. *See* Selecting objects

Setting display options 112

- General view defaults 119
- Saving and restoring user customized themes 117
- Setting Bundle Nets display options 115
- Setting Default Light or Dark Themes 117
- Setting Device view defaults 113
- Setting I/O view defaults 114
- Setting look and feel options 119
- Setting Software update check 120

Status Bar 107

- Clock 107
- Coordinates field 107
- Current mode field 107
- Information message field 107
- Java memory consumption gauge 107

W

WASSO analysis 161

- Running WASSO analysis 161
- Viewing WASSO analysis results 162

Workspace 102

- Closing Workspace views 103
- Common commands 251
- Opening Workspace views 103
- Splitting the Workspace 104
- Views 102

World view 106

X

Xilinx Technical Support case files to include 25

XilinxUpdate

- Automatic check for updates 120